



Altera Transceiver PHY IP Core

User Guide



101 Innovation Drive
San Jose, CA 95134
www.altera.com

UG-01080-1.6

Document last updated for Altera Complete Design Suite version:
Document publication date:

11.1 SP2
March 2012



© 2012 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.



Chapter 1. Introduction

PCS	1-2
PMA	1-3
Reset Controller	1-3
Avalon-MM PHY Management	1-3
Running a Simulation Testbench	1-4
Unsupported Features	1-5

Chapter 2. Getting Started

Installation and Licensing	2-1
Design Flows	2-1
MegaWizard Plug-In Manager Flow	2-2
Specifying Parameters	2-2
Simulate the IP Core	2-4

Chapter 3. 10GBASE-R PHY IP Core

Release Information	3-2
Device Family Support	3-3
Performance and Resource Utilization	3-3
Stratix IV Devices	3-3
Stratix V Devices	3-4
Parameter Settings	3-4
General Options	3-4
Analog Options	3-5
Stratix IV Devices	3-5
Stratix V Devices	3-6
Interfaces	3-11
Ports	3-11
SDR XGMII TX Interface	3-12
SDR XGMII RX Interface	3-13
Status Interface	3-14
Clocks, Reset, and Powerdown	3-14
Serial Interface	3-17
Register Interface	3-17
Register Descriptions	3-18
Dynamic Reconfiguration	3-20
Dynamic Reconfiguration for Stratix IV Devices	3-20
Dynamic Reconfiguration for Stratix V Devices	3-21
TimeQuest Timing Constraints	3-22
Simulation Files and Example Testbench	3-24

Chapter 4. XAUI PHY IP Core

Release Information	4-2
Device Family Support	4-2
Performance and Resource Utilization	4-3
Parameter Settings	4-3
Analog Options	4-4
Arria II GX, Cyclone IV GX, HardCopy IV and Stratix IV Devices	4-4

Stratix V Devices	4-5
Advanced Options	4-9
Configurations	4-10
Interfaces	4-10
Ports	4-11
SDR XGMII TX Interface	4-12
SDR XGMII RX Interface	4-14
Transceiver Serial Data Interface	4-14
Clocks, Reset, and Powerdown	4-15
PMA Channel Controller	4-16
PMA Control and Status Interface Signals–Soft IP Implementation (Optional)	4-16
PMA Control and Status Interface Signals–Hard IP Implementation (Optional)	4-17
Registers	4-19
Register Descriptions	4-19
Dynamic Reconfiguration	4-23
Reconfiguration for Arria II GX, Cyclone IV GX, HardCopy IV GX, and Stratix IV GX Devices ...	4-23
Dynamic Reconfiguration for Stratix V Devices	4-24
Simulation Files and Example Testbench	4-24

Chapter 5. Interlaken PHY IP Core

Device Family Support	5-2
Parameter Settings	5-2
General Options	5-2
Advanced Options	5-3
Analog Settings	5-4
Interfaces	5-8
Ports	5-8
Avalon-ST TX Interface	5-9
Avalon-ST RX Interface	5-10
PLL Interface	5-12
TX and RX Serial Interface	5-12
Optional Clocks for Deskew	5-12
Registers	5-13
Register Descriptions	5-13
Transceiver Reconfiguration	5-15
TimeQuest Timing Constraints	5-16
Simulation Files and Example Testbench	5-16

Chapter 6. PHY IP Core for PCI Express (PIPE)

Device Family Support	6-1
Resource Utilization	6-2
Parameter Settings	6-2
General Options	6-2
Analog Options	6-3
Interfaces	6-8
Ports	6-8
Avalon-ST TX Input Data from the PHYMAC	6-9
Avalon-ST RX Output Data to the PHYMAC	6-9
PIPE Interface	6-10
Transceiver Serial Interface	6-12
Optional Status Interface	6-12
Registers	6-13

Register Descriptions	6-14
Dynamic Reconfiguration	6-17
Simulation Files and Example Testbench	6-18

Chapter 7. Custom PHY IP Core

Device Family Support	7-2
Performance and Resource Utilization	7-2
Parameter Settings	7-3
General Options	7-3
Word Alignment	7-6
Rate Match FIFO	7-7
8B/10B Encoder and Decoder	7-8
Byte Ordering	7-8
PLL Reconfiguration	7-9
Analog Options	7-10
Presets for Ethernet	7-14
Interfaces	7-15
Ports	7-15
Avalon-ST TX Input Data from the MAC	7-17
Avalon-ST RX Output Data to the MAC	7-17
Clock Interface	7-18
Transceiver Serial Data Interface	7-18
Status Signals (Optional)	7-18
Reset Control and Status (Optional)	7-19
Register Interface	7-20
Register Descriptions	7-21
Dynamic Reconfiguration	7-23
Simulation Files and Example Testbench	7-24

Chapter 8. Low Latency PHY IP Core

Device Family Support	8-1
Performance and Resource Utilization	8-2
Parameter Settings	8-3
General Options	8-3
Additional Options	8-5
PLL Reconfiguration Options	8-6
Analog Options	8-8
Interfaces	8-12
Ports	8-12
Avalon-ST TX and RX Data Interface to the FPGA Fabric	8-13
Serial Data Interface	8-13
Optional Status Interface	8-14
Clock Interface	8-14
Reset Control and Status (Optional)	8-14
Register Interface	8-15
Register Descriptions	8-16
Dynamic Reconfiguration	8-17
Simulation Files and Example Testbench	8-17

Chapter 9. Deterministic Latency PHY IP Core

Auto-Negotiation	9-2
Achieving Deterministic Latency	9-3
Delay Estimation Logic	9-3

Delay Numbers	9-4
Device Family Support	9-5
Parameter Settings	9-6
General Options	9-6
Additional Options	9-8
Analog Options	9-9
Interfaces	9-14
Ports	9-14
Avalon-ST TX Input Data from the MAC	9-16
Avalon-ST RX Output Data to the MAC	9-16
Clock Interface	9-16
Transceiver Serial Data Interface	9-17
TX and RX Status Signals	9-17
Optional Reset Control and Status	9-18
Register Interface	9-18
Register Descriptions	9-20
Dynamic Reconfiguration	9-22
Channel Placement and Utilization	9-23
Simulation Files and Example Testbench	9-24

Chapter 10. Transceiver Reconfiguration Controller

System Overview	10-3
Device Family Support	10-4
Performance and Resource Utilization	10-5
Parameter Settings	10-5
Interfaces	10-7
MIF Reconfiguration Management Avalon-MM Master Interface	10-7
Transceiver Reconfiguration Interface	10-8
Reconfiguration Interface Management Interface	10-8
Reconfiguration Controller Memory Map	10-9
Transceiver Calibration Functions	10-10
Offset Cancellation	10-10
Duty Cycle Calibration	10-10
Auxiliary Transmit (ATX) PLL Calibration	10-10
PMA Analog Controls	10-11
EyeQ	10-12
DFE	10-14
AEQ	10-16
ATX PLL Calibration	10-17
PLL Reconfiguration	10-18
Channel and PLL Reconfiguration	10-21
Channel Reconfiguration	10-21
PLL Reconfiguration	10-22
Streamer Module	10-22
Mode 0 Streaming a MIF for Reconfiguration	10-24
Mode 1 Avalon-MM Direct Writes for Reconfiguration	10-25
Stratix V MIF	10-25
MIF Format	10-25
Procedures for Reconfiguration	10-27
Changing Transceiver Settings Using Register-Based Reconfiguration	10-27
Register-Based Write	10-27
Register-Based Read	10-27
Changing Transceiver Settings Using Streamer-Based Reconfiguration	10-28
Streamer Based Reconfiguration	10-28

Direct Write Reconfiguration	10-29
Understanding Logical Channel Numbering	10-31
Two PHY IP Core Instances Each with Four Bonded Channels	10-34
One PHY IP Core Instance with Eight Bonded Channels	10-35
Two PHY IP Core Instances Each with Non-Bonded Channels	10-36
Reconfiguration Controller to PHY IP Connectivity	10-37
Merging TX PLLs In Multiple Transceiver PHY Instances	10-38
Loopback Modes	10-39

Chapter 11. Migrating from Stratix IV to Stratix V Devices

Dynamic Reconfiguration of Transceivers	11-2
Dynamic Reconfiguration for Stratix V Transceivers	11-2
Dynamic Reconfiguration for Stratix IV Transceivers	11-2
XAUI PHY	11-3
Parameter Differences	11-3
Port Differences	11-4
PHY IP Core for PCI Express PHY (PIPE)	11-6
Parameter Differences	11-6
Port Differences	11-7
Custom PHY	11-10
Parameter Differences	11-10
Port Differences	11-11

Additional Information

Revision History	Info-1
How to Contact Altera	Info-10
Typographic Conventions	Info-10

The *Altera® Transceiver PHY IP Core User Guide* describes the following protocol-specific PHYs:

- 10GBASE-R PHY IP Core
- XAUI PHY IP Core
- Interlaken PHY IP Core
- PHY IP Core for PCI Express (PIPE)
- Custom PHY IP Core
- Low Latency PHY IP Core
- Deterministic Latency PHY IP Core

The protocol-specific PHYs automatically configure settings for the physical coding sublayer (PCS) module, leaving a small number of parameters in the physical media attachment (PMA) module for you to configure. You can use the Custom PHY or Low Latency PHY for applications that require more flexible settings. The design of all of these PHYs is modular and uses standard interfaces. All PHYs include an Avalon® Memory-Mapped (Avalon-MM) interface to access control and status registers and an Avalon Streaming (Avalon-ST) interface to connect to the MAC for data transfer. The control and status registers store device-dependent information about the PCS and PMA modules. You can access this device-dependent information using the device-independent Avalon-MM interface, reducing overall complexity of your design and the number of device-dependent signals that you must expose in your top-level module.



For more information about the Avalon-MM and Avalon-ST protocols, including timing diagrams, refer to the [Avalon Interface Specifications](#).

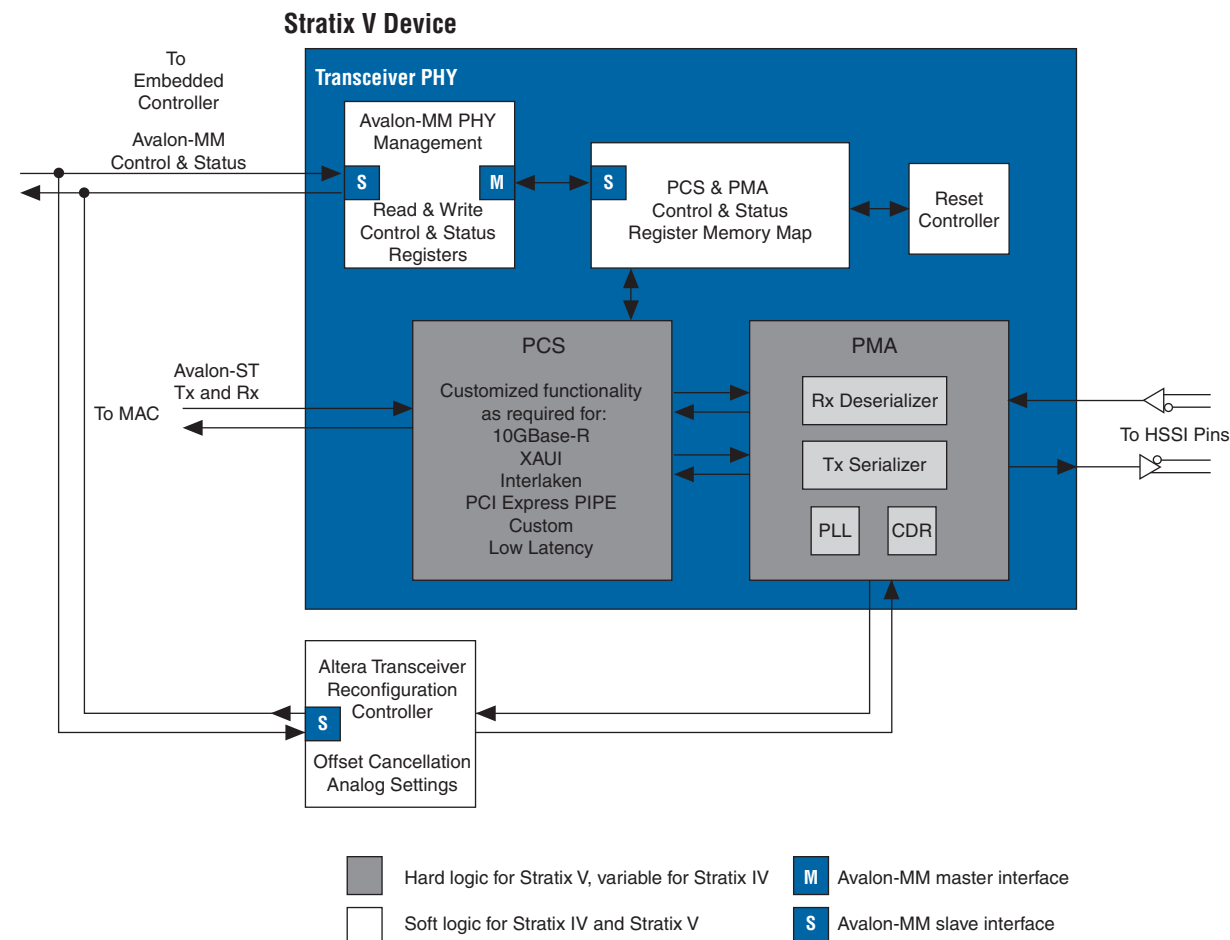
Table 1–1 shows hard and soft implementation support for these transceiver PHY IP cores in Stratix® V devices. Typically, the PCS and PMA are implemented as hard logic, saving FPGA resources and reducing the complexity of verification. In some cases, the PCS is also available in soft logic as **Table 1–1** indicates.

Table 1–1. Stratix V GX Support for Protocol Specific PHY IP Cores

PHY Protocol	Soft PCS	Hard PCS	Hard PMA
10GBASE-R	No	Yes	Yes
XAUI	Yes	No	Yes
Interlaken	No	Yes	Yes
PCI Express Gen1 and Gen2	No	Yes	Yes
Custom PHY	No	Yes	Yes
Low latency PHY	No	Yes	Yes
Deterministic Latency PHY	No	Yes	Yes

Figure 1–1 illustrates the top level modules that comprise the transceiver PHY IP cores. In addition, Figure 1–1 shows the Altera Transceiver Reconfiguration Controller IP core that is instantiated separately in Stratix V devices.

Figure 1–1. Altera Modular PHY Design



The following sections provide a brief introduction to each of the modules illustrated in Figure 1–1.

PCS

The PCS implements part of the physical layer specification for networking protocols. Depending upon the protocol that you choose, the PCS may include many different functions. Some of the most commonly included functions are: 8B/10B, 64B/66B, or 64B/67B encoding and decoding, rate matching and clock compensation, scrambling and de-scrambling, word alignment, phase compensation, error monitoring, and gearbox.

PMA

The PMA receives and transmits differential serial data on the device external pins. The transmit (TX) channel supports programmable pre-emphasis and programmable output differential voltage (V_{OD}). It converts parallel input data streams to serial data. The receive (RX) channel supports offset cancellation to correct for process variation and programmable equalization. It converts serial data to parallel data for processing in the PCS. The PMA also includes a clock data recovery (CDR) module with separate CDR logic for each RX channel.

Reset Controller

A transceiver reset controller is included as part of each PHY IP core. This embedded reset controller ensures reliable transceiver link initialization. The reset controller initializes the both the TX and RX channels. You can disable the automatic reset controller in the Custom and Low Latency Transceiver PHYs. If you do disable the embedded reset controller, the powerdown, analog and digital reset signals for both the TX and RX channels are top-level ports of the transceiver PHY. You can use these signals to design a custom reset sequence.

To accommodate different reset requirements for different transceivers in your design, instantiate multiple instances of a PHY IP core. For example, if your design includes 20 channels of the Custom PHY IP core with 12 channels running a custom protocol using the automatic reset controller and 8 channels requiring manual control of RX reset, instantiate 2 instances of the Custom PHY IP core and customize one to use automatic mode and the other to use your own reset logic. For more information, refer to “Enable embedded reset control” in [Table 7-3 on page 7-3](#).



For more information about reset in Stratix V devices, refer to [Transceiver Reset Control in Stratix V Devices](#) in volume 3 of the *Stratix V Device Handbook*, for Stratix IV devices, refer to [Reset Control and Power Down](#) in volume 4 of the *Stratix IV Device Handbook*.

Avalon-MM PHY Management

You can use the Avalon-MM PHY Management module to read and write the control and status registers in the PCS and PMA. This module includes both Avalon-MM master and slave ports and acts as a bridge. It transfers commands received from an embedded controller on its slave port to its master port. The Avalon-MM PHY management master interface connects the Avalon-MM slave ports of PCS and PMA registers and the Transceiver Reconfiguration module, allowing you to manage these Avalon-MM slave components through a simple, standard interface. (Refer to [Figure 1-1 on page 1-2](#).)

Running a Simulation Testbench

When you generate your transceiver PHY IP core, the Quartus II software generates the HDL files that define your parameterized IP core. In addition, the Quartus II software generates an example Tcl script to compile and simulate your design in ModelSim. Figure 1-2 illustrates the directory structure for the generated files.

Figure 1-2. Directory Structure for Generated Files

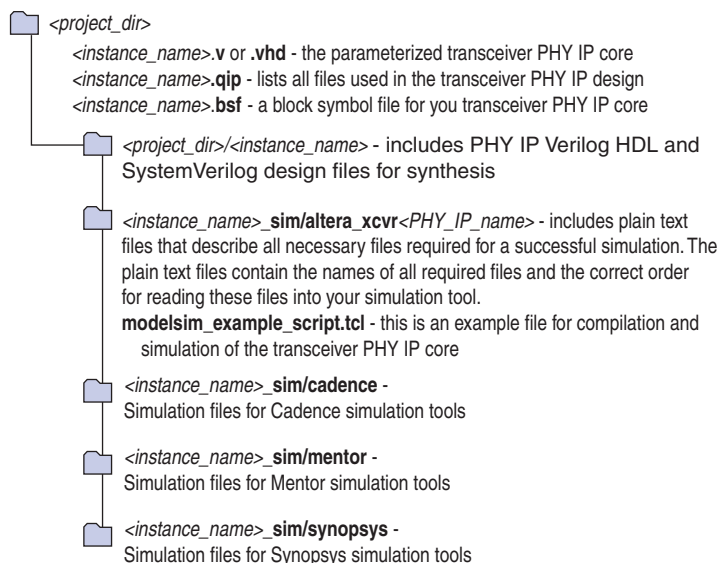


Table 1-2 describes the key files and directories for the parameterized transceiver PHY IP core and the simulation environment which are in clear text.

Table 1-2. Generated Files (Part 1 of 2)

File Name	Description
<project_dir>	The top-level project directory.
<instance_name>.v or .vhd	The top-level design file.
<instance_name>.qip	A list of all files necessary for Quartus II compilation.
<instance_name>.bsf	A Block Symbol File (.bsf) for your transceiver PHY.
<project_dir>/<instance_name>/	The directory that stores the HDL files that define the protocol-specific PHY IP core. These files are used for synthesis.
<instance_name>_phy_assignments.qip	Includes an example of the PLL_TYPE assignment statement required to specify the PLL type for each PLL in the design. The available types are clock multiplier unit (CMU) and auxiliary transmit (ATX).
<project_dir>/<instance_name>_sim/ altera_xcvr_<PHY_IP_name>/	The simulation directory.
modelsim_example_script.tcl	The example Tcl script to compile and simulate the parameterized transceiver PHY IP core. You must edit this script to include the following information: ■ The simulation language ■ The top-level transceiver PHY variation name ■ The name of your testbench

Table 1–2. Generated Files (Part 2 of 2)

File Name	Description
<code><project_dir>/<instance_name>_sim/cadence</code>	Simulation file for Cadence simulation tools.
<code><project_dir>/<instance_name>_sim/mentor</code>	Simulation file for Mentor simulation tools.
<code><project_dir>/<instance_name>_sim/synopsys</code>	Simulation file for Synopsys simulation tools.

The Verilog and VHDL transceiver PHY IP cores have been tested with the following simulators:

- ModelSim SE
- Synopsys VCS MX
- Cadence NCSim

If you select VHDL for your transceiver PHY, only the wrapper generated by the Quartus II software is in VHDL. All the underlying files are written in Verilog or System Verilog. To enable simulation using a VHDL-only ModelSim license, the underlying Verilog and System Verilog files for the transceiver PHY are encrypted so that they can be used with the top-level VHDL wrapper without using a mixed-language simulator.



For more information about simulating with ModelSim, refer to the *Mentor Graphics ModelSim Support* chapter in volume 3 of the *Quartus II Handbook*



The transceiver PHY IP cores do not support the NativeLink feature in the Quartus II software.

Unsupported Features

The protocol-specific PHYs are not supported in SOPC Builder or Qsys in the current release.

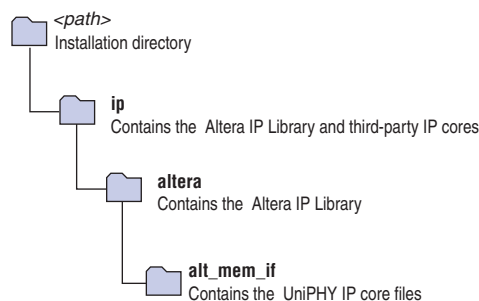
This chapter provides a general overview of the Altera IP core design flow to help you quickly get started with any Altera IP core. The Altera IP Library is installed as part of the Quartus II installation process. You can select and parameterize any Altera IP core from the library. Altera provides an integrated parameter editor that allows you to customize IP cores to support a wide variety of applications. The parameter editor guides you through the setting of parameter values and selection of optional ports. The following sections describe the general design flow and use of Altera IP cores.

Installation and Licensing

The Altera IP Library is distributed with the Quartus II software and downloadable from the Altera website (www.altera.com).

Figure 2–1 shows the directory structure after you install an Altera IP core, where *<path>* is the installation directory. The default installation directory on Windows is `C:\altera\<version number>`; on Linux it is `/opt/altera<version number>`.

Figure 2–1. IP core Directory Structure



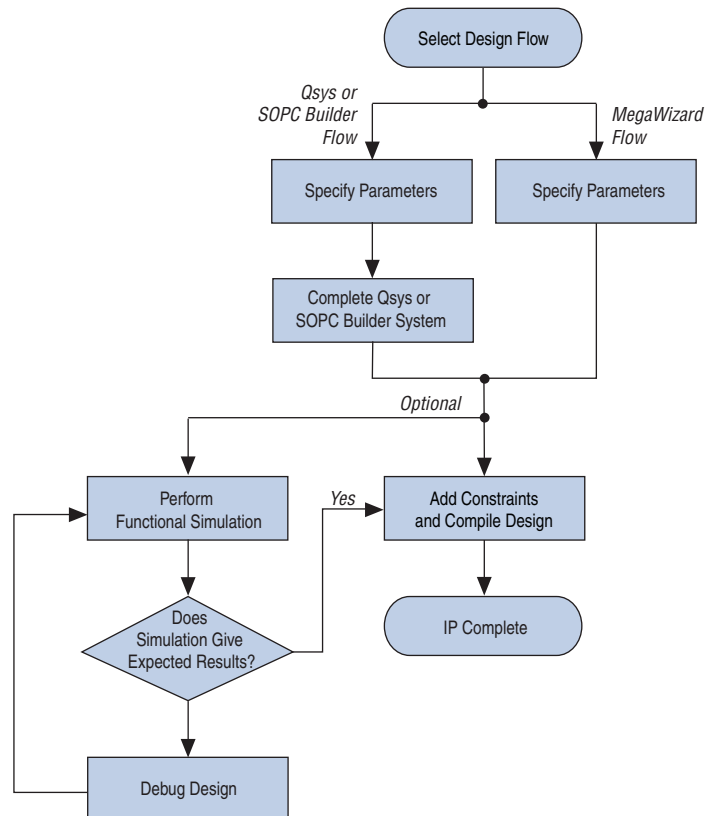
You can evaluate an IP core in simulation and in hardware until you are satisfied with its functionality and performance. Some IP cores require that you purchase a license for the IP core when you want to take your design to production. After you purchase a license for an Altera IP core, you can request a license file from the [Altera Licensing](#) page of the Altera website and install the license on your computer. For additional information, refer to [Altera Software Installation and Licensing](#).

Design Flows

You can use the following flow(s) to parameterize Altera IP cores:

■ MegaWizard™ Plug-In Manager Flow

Figure 2–2. Design Flows (1)



Note to Figure 2–2:

(1) Altera IP cores may or may not support the Qsys and SOPC Builder design flows.

The MegaWizard Plug-In Manager flow offers the following advantages:

- Allows you to parameterize an IP core variant and instantiate into an existing design
- For some IP cores, this flow generates a complete example design and testbench

MegaWizard Plug-In Manager Flow

The MegaWizard Plug-In Manager flow allows you to customize your IP core and manually integrate the function into your design.

Specifying Parameters

To specify IP core parameters with the MegaWizard Plug-In Manager, follow these steps:

1. Create a Quartus II project using the **New Project Wizard** available from the File menu.

2. In the Quartus II software, launch the **MegaWizard Plug-in Manager** from the Tools menu, and follow the prompts in the MegaWizard Plug-In Manager interface to create or edit a custom IP core variation.
3. To select a specific Altera IP core, click the IP core in the **Installed Plug-Ins** list in the MegaWizard Plug-In Manager.
4. Specify the parameters on the **Parameter Settings** pages. For detailed explanations of these parameters, refer to the “*Parameter Settings*” chapter in this document or the “*Documentation*” button in the MegaWizard parameter editor.



Some IP cores provide preset parameters for specific applications. If you wish to use preset parameters, click the arrow to expand the **Presets** list, select the desired preset, and then click **Apply**. To modify preset settings, in a text editor modify the `<installation_directory>/ip/altera/alt_mem_if_interfaces/alt_mem_if_<memory_protocol>_emif/alt_mem_if_<memory_protocol>_mem_model.qprs` file.

5. If the IP core provides a simulation model, specify appropriate options in the wizard to generate a simulation model.



Altera IP supports a variety of simulation models, including simulation-specific IP functional simulation models and encrypted RTL models, and plain text RTL models. These are all cycle-accurate models. The models allow for fast functional simulation of your IP core instance using industry-standard VHDL or Verilog HDL simulators. For some cores, only the plain text RTL model is generated, and you can simulate that model.



For more information about functional simulation models for Altera IP cores, refer to *Simulating Altera Designs* in volume 3 of the *Quartus II Handbook*.



Use the simulation models only for simulation and not for synthesis or any other purposes. Using these models for synthesis creates a nonfunctional design.

6. If the parameter editor includes **EDA** and **Summary** tabs, follow these steps:
 - a. Some third-party synthesis tools can use a netlist that contains the structure of an IP core but no detailed logic to optimize timing and performance of the design containing it. To use this feature if your synthesis tool and IP core support it, turn on **Generate netlist**.
 - b. On the **Summary** tab, if available, select the files you want to generate. A gray checkmark indicates a file that is automatically generated. All other files are optional.



If file selection is supported for your IP core, after you generate the core, a generation report (`<variation name>.html`) appears in your project directory. This file contains information about the generated files.

7. Click the **Finish** button, the parameter editor generates the top-level HDL code for your IP core, and a simulation directory which includes files for simulation.



The **Finish** button may be unavailable until all parameterization errors listed in the messages window are corrected.

8. Click **Yes** if you are prompted to add the Quartus II IP File (.qip) to the current Quartus II project. You can also turn on **Automatically add Quartus II IP Files to all projects**.

You can now integrate your custom IP core instance in your design, simulate, and compile. While integrating your IP core instance into your design, you must make appropriate pin assignments. You can create a virtual pin to avoid making specific pin assignments for top-level signals while you are simulating and not ready to map the design to hardware.

For some IP cores, the generation process also creates complete example designs. An example design for hardware testing is located in the `<variation_name>_example_design/example_project/` directory. An example design for RTL simulation is located in the `<variation_name>_example_design/simulation/` directory.



For information about the Quartus II software, including virtual pins and the MegaWizard Plug-In Manager, refer to [Quartus II Help](#).

Simulate the IP Core

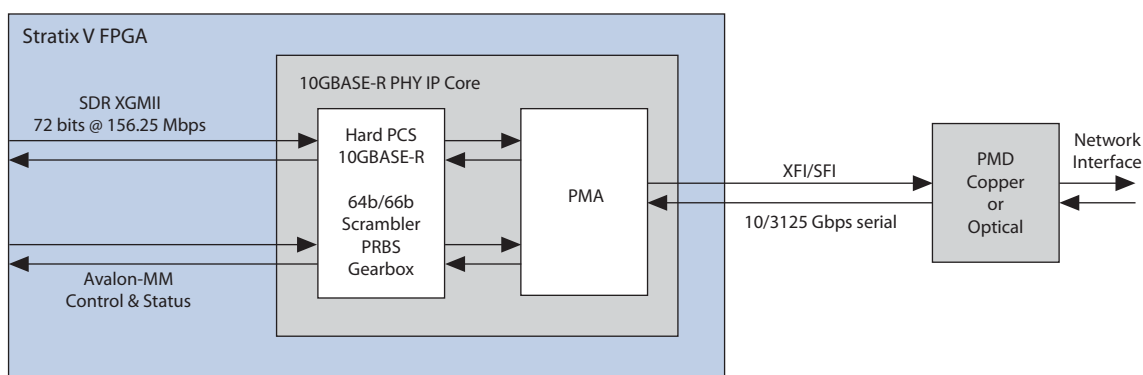
You can simulate your IP core variation with the functional simulation model and the testbench or example design generated with your IP core. The functional simulation model and testbench files are generated in a project subdirectory. This directory may also include scripts to compile and run the testbench.

For a complete list of models or libraries required to simulate your IP core, refer to the scripts provided with the testbench.

For more information about simulating Altera IP cores, refer to [Simulating Altera Designs](#) in volume 3 of the *Quartus II Handbook*.

The Altera 10GBASE-R PHY IP core implements the functionality described in [IEEE 802.3 Clause 49](#). It delivers serialized data to an optical module that drives optical fiber at a line rate of 10.3125 Gbps. In a multi-channel implementation of 10GBASE-R, each channel of the 10GBASE-R PHY IP core operates independently. [Figure 3–1](#) shows the 10GBASE-R PHY IP core available for Stratix V devices. Both the PCS and PMA of the 10GBASE-R PHY are implemented as hard IP blocks in Stratix V devices, saving FPGA resources.

Figure 3–1. 10GBASE-R PHY with Hard PCS with PMA in Stratix V Devices



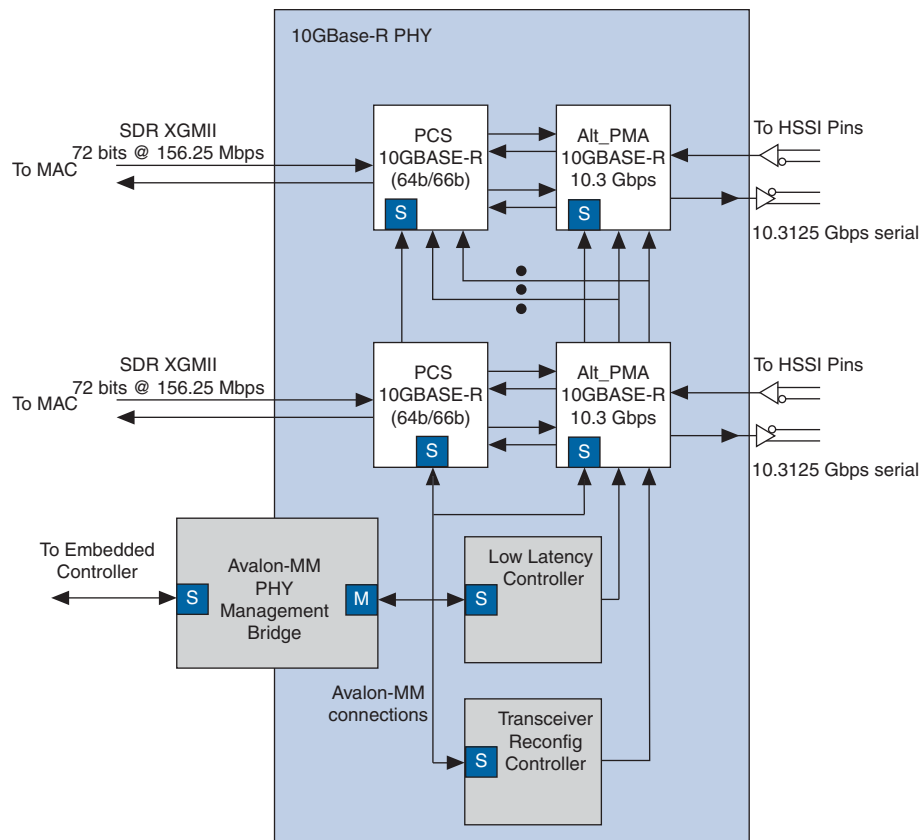
- For a 10-Gbps Ethernet solution that includes both the Ethernet MAC and the 10GBASE-R PHY, refer to the [10-Gbps Ethernet MAC MegaCore Function User Guide](#).
- For more detailed information about the 10GBASE-R transceiver channel datapath, clocking, and channel placement, refer to the “10GBASE-R” section in the [Transceiver Protocol Configurations in Stratix V Devices](#) chapter of the *Stratix V Device Handbook*.

[Figure 3–2](#) illustrates a multiple 10 GbE channel IP core in a Stratix IV GT device. To achieve higher bandwidths, you can instantiate multiple channels. The PCS is available in soft logic for Stratix IV GT devices; it connects to a separately instantiated hard PMA. The PCS connects to an Ethernet MAC via single data rate (SDR) XGMII running at 156.25 megabits per second (Mbps) and transmits data to a 10 Gbps transceiver PMA running at 10.3125 gigabits per second (Gbps) in a Stratix IV GT device.

To make most effective use of this soft PCS and PMA configuration for Stratix IV GT devices, you can group up to four channels in a single quad and control their functionality using one Avalon-MM PHY management bridge, transceiver reconfiguration module, and low latency controller. As Figure 3-2 illustrates, the Avalon-MM bridge Avalon-MM master port connects to the Avalon-MM slave port of the transceiver reconfiguration and low latency controller modules so that you can update analog settings using the standard Avalon-MM interface.

This configuration does not require that all four channels in a quad run the 10GBASE-R protocol.

Figure 3-2. Complete 10GBASE-R PHY Design in Stratix IV GT Device



Release Information

Table 3-1 provides information about this release of the 10GBASE-R PHY IP core.

Table 3-1. 10GBASE-R Release Information (Part 1 of 2)

Item	Description
Version	11.1
Release Date	November 2011
Ordering Codes ⁽¹⁾	IP-10GBASERPCS (primary) IPR-10GBASERPCS (renewal)
Product ID	00D7

Table 3–1. 10GBASE-R Release Information (Part 2 of 2)

Item	Description
Vendor ID	6AF7

Note to Table 3–1:

(1) No ordering codes or license files are required for Stratix V devices.

Device Family Support

IP cores provide either final or preliminary support for target Altera device families. These terms have the following definitions:

- *Final support*—Verified with final timing models for this device.
- *Preliminary support*—Verified with preliminary timing models for this device.

Table 3–2 shows the level of support offered by the 10GBASE-R IP core for Altera device families.

Table 3–2. Device Family Support

Device Family	Support
Stratix IV GT devices—Soft PCS and PMA	Final
Stratix V devices—Hard PCS and PMA	Preliminary—C2 Speed Grade
Other device families	No support



For speed grade information, refer to “Transceiver Performance Specifications” in the *DC and Switching Characteristics* chapter in the *Stratix IV Handbook* for Stratix IV devices or *DC and Switching Characteristics for Stratix V Devices* in the *Stratix V Device Handbook* for Stratix V devices.

Performance and Resource Utilization

This section provides information on performance and resource utilization for Stratix IV and Stratix V devices.

Stratix IV Devices

Table 3–3 shows the typical expected device resource utilization for duplex channels using the current version of the Quartus II software targeting a Stratix IV GT device. The numbers of combinational ALUTs, logic registers, and memory bits are rounded to the nearest 100.

Table 3–3. 10GBASE-R PHY Performance and Resource Utilization—Stratix IV GT Device

Channels	Combinational ALUTs	Logic Registers (Bits)	Memory Bits
1	5200	4100	4700
4	15600	1300	18800
10	38100	32100	47500

Stratix V Devices

For Stratix V devices, the PCS and PMA are both implemented in hard logic; the 10GBASE-R transceiver PHY requires less than 1% of FPGA resources.

Table 3-4 lists the total latency for an Ethernet packet with a 9600 byte payload and an inter-packet gap of 12 characters. The latency includes the number of cycles to transmit the payload from the TX XGMII interface, through the TX PCS and PMA, looping back through the RX PMA and PCS to the RX XGMII interface. (Figure 3-5 on page 3-16 illustrates this datapath.)

Table 3-4. Latency

PPM Difference	Cycles
0 PPM	35
-200 PPM	35
+200 PPM	42



It latency is critical, Altera recommends designing your own soft 10GBASE-R PCS and connecting to the “Low Latency PHY IP Core” .

Parameter Settings

To configure the 10GBASE-R PHY IP core in the parameter editor, click **Installed Plug-Ins > Interfaces > Ethernet > 10GBASE-R PHY v11.1**. The 10GBASE-R PHY IP core is available for the **Stratix IV** or **Stratix V** device family.

General Options

This section describes the 10GBASE-R PHY parameters, which you can set using the parameter editor. Table 3-5 lists the settings available on **General Options** tab.

Table 3-5. General Options (Part 1 of 2)

Name	Value	Description
General Options		
Device family	Stratix IV GT Stratix V	Specifies the target device.
Number of channels	1–32	The total number of 10GBASE-R PHY channels.
Mode of operation	Duplex TX only RX only	Stratix V devices allow duplex, TX, or RX mode. Stratix IV GX devices only support duplex mode.

Table 3–5. General Options (Part 2 of 2)

Name	Value	Description
PLL type	CMU ATX	You can select either the CMU or ATX PLL. The CMU PLL has a larger frequency range than the ATX PLL. The ATX PLL is designed to improve jitter performance and achieves lower channel-to-channel skew; however, it supports a narrower range of data rates and reference clock frequencies. Another advantage of the ATX PLL is that it does not use a transceiver channel, while the CMU PLL does. Because the CMU PLL is more versatile, it is specified as the default setting. An informational message displays in the message pane telling you whether the chosen settings for Data rate and Input clock frequency are legal for the CMU PLL, or for both the CMU and ATX PLLs.
Reference Clock Frequency	322.265625 MHz 644.53125 MHz	Stratix V devices support both frequencies. Stratix IV GX devices only support 644.53125 MHz .
Additional Options		
Use external PMA control and reconfig	On/Off	For Stratix IV devices, if you turn this option on, the PMA controller and reconfiguration block are external, rather than included 10GBASE-R PHY IP core, allowing you to use the same PMA controller and reconfiguration IP cores for other protocols in the same transceiver quad. When you turn this option On , the <code>cal_blk_powerdown</code> (0x021) and <code>pma_tx_pll_is_locked</code> (0x022) registers are available.
Enable additional control and status pins	On/Off	If you turn this option on, the following 2 signals are brought out to the top level of the IP core to facilitate debugging: <code>hi_ber</code> and <code>block_lock</code> .
Enable rx_recovered_clk pin	On/Off	When you turn this option on, the RX recovered clock signal is an output signal.
Starting channel number	0–96	For Stratix IV devices, specifies the starting channel number. Must be 0 or a multiple of 4. You only need to set this parameter if you are using external PMA and reconfiguration modules. This option is only necessary for Stratix IV devices.

Analog Options

The following sections explain how to specify analog options for Stratix IV and Stratix V devices.

Stratix IV Devices

For Stratix IV devices, you specify analog options on the **Analog Options** tab. [Table 3–6](#) describes these options.

Table 3–6. PMA Analog Options (Part 1 of 2)

Name	Value	Description
Transmitter termination resistance	OCT_85_OHMS OCT_100_OHMS OCT_120_OHMS OCT_150_OHMS	Indicates the value of the termination resistor for the transmitter.
Transmitter VOD control setting	0–7	Sets V_{OD} for the various TX buffers.

Table 3-6. PMA Analog Options (Part 2 of 2)

Name	Value	Description
Pre-emphasis pre-tap setting	0-7	Sets the amount of pre-emphasis on the TX buffer.
Invert the pre-emphasis pre-tap polarity setting	On Off	Determines whether or not the pre-emphasis control signal for the pre-tap is inverted. If you turn this option on, the pre-emphasis control signal is inverted.
Pre-emphasis first post-tap setting	0-15	Sets the amount of pre-emphasis for the 1st post-tap.
Pre-emphasis second post-tap setting	0-7	Sets the amount of pre-emphasis for the 2nd post-tap.
Invert the pre-emphasis second post-tap polarity	On Off	Determines whether or not the pre-emphasis control signal for the second post-tap is inverted. If you turn this option on, the pre-emphasis control signal is inverted.
Receiver common mode voltage	Tri-State 0.82V 1.1v	Specifies the RX common mode voltage.
Receiver termination resistance	OCT_85_OHMS OCT_100_OHMS OCT_120_OHMS OCT_150_OHMS	Indicates the value of the termination resistor for the receiver.
Receiver DC	0-4	Sets the equalization DC gain using one of the following settings: ■ 0-0 dB ■ 1-3 dB ■ 2-6 dB ■ 3-9 dB ■ 4-12 dB
Receiver static equalizer setting:	0-15	This option sets the equalizer control settings. The equalizer uses a pass band filter. Specifying a low value passes low frequencies. Specifying a high value passes high frequencies.

Stratix V Devices

You specify the analog parameters for Stratix V devices using the Quartus II Assignment Editor, the Pin Planner, or through the Quartus II Settings File (.qsf). The default values for analog options fall into three categories:

- *Global*— These parameters have default values that are independent of other parameter settings.
- *Computed*—These parameters have an initial default value that is recomputed based on other parameter settings.
- *Proxy*—These parameters have default values that are place holders. The Quartus II software selects these initial default values based on your design; however, Altera recommends that you replace these defaults with values that match your electrical board specification.

Table 3-7 lists the analog parameters for Stratix V devices whose original values are place holders for the values that match your electrical board specification. In Table 3-7, the default value of an analog parameter is shown in **bold** type.

Table 3-7. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 2)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_IO_PIN_TERMINATION	Transceiver I/O Pin Termination	Specifies the intended on-chip termination value for the specified transceiver pin. Use External Resistor if you intend to use off-chip termination.	85_OHMS 100_OHMS 120_OHMS 150_OHMS EXTERNAL_RESISTOR	Pin
XCVR_REFCLK_PIN_TERMINATION	Transceiver Dedicated Refclk Pin Termination	Specifies the intended termination value for the specified refclk pin.	DC_COUPLING_INTERNAL_100_OHM DC_COUPLING_EXTERNAL_RESISTOR AC_COUPLING	Pin
XCVR_RX_BYPASS_EQ_STAGES_234	Receiver Equalizer Stage 2, 3, 4 Bypass	Bypass continuous time equalizer stages 2, 3, and 4 to save power. This setting eliminates significant AC gain on the equalizer and is appropriate for chip-to-chip short range communication on a PCB.	ALL_STAGES_ENABLED BYPASS_STAGES	Pin
XCVR_TX_SLEW_RATE_CTRL	Transmitter Slew Rate Control	Specifies the slew rate of the output signal. The valid values span from the slowest rate to fastest rate with 1 representing the slowest rate.	1-5	Pin

Table 3–7. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 2)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_VCCA_VOLTAGE	VCCA_GXB Voltage	Configure the VCCA_GXB voltage for a GXB I/O pin by specifying the intended VCCA_GXB voltage for a GXB I/O pin. If you do not make this assignment the compiler automatically sets the correct VCCA_GXB voltage depending on the configured data rate, as follows: ■ Data rate <= 6.5 Gbps: 2_5V ■ Data rate > 6.5 Gbps: 3_0V or 3_3V for Stratix V ES silicon	2_5V 3_0V	Pin
XCVR_VCCR_VCCT_VOLTAGE	VCCR_GXB VCCT_GXB Voltage	Configure the VCCR_GXB and VCCT_GXB voltage for an GXB I/O pin by specifying the intended supply voltages for a GXB I/O pin. If your design uses decision feedback equalization (DFE) or adaptive equalization (AEQ), you must set this parameter to 1.0V. Otherwise, if you do not make this assignment the compiler automatically sets the correct VCCR_GXB and VCCT_GXB voltage depending on the configured data rate as follows: ■ Data rate <= 6.5 Gbps: 0_85V ■ Data rate > 6.5 Gbps: 1_0V	0_85V 1_0V	Pin

Table 3–8 lists the analog parameters with *global* or *computed* default values. You may want to optimize some of these settings. In Table 3–8, the default value is shown in **bold** type. For computed analog parameters, the default value listed is for the initial setting, not the recomputed setting.

Table 3–8. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 3)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
Analog Parameters with Global Default Value				
CDR_BANDWIDTH_PRESET	CDR Bandwidth Preset	Specifies the CDR bandwidth preset setting.	Auto Low Medium High	PLL instance
PLL_BANDWIDTH_PRESET	PLL Bandwidth Preset	Specifies the PLL bandwidth preset setting	Auto Low Medium High	PLL instance
XCVR_RX_DC_GAIN	Receiver Buffer DC Gain Control	Controls the amount of a stage receive-buffer DC gain.	0–4	Pin

Table 3–8. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 3)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_RX_LINEAR_EQUALIZER_CONTROL	Receiver Linear Equalizer Control	Static control for the continuous time equalizer in the receiver buffer. The equalizer has 16 distinct settings from 0–15 corresponding to the increasing AC gain.	1–16	Pin
Analog Parameter with Computed Default Value				
XCVR_RX_COMMON_MODE_VOLTAGE	Receiver Buffer Common Mode Voltage	Receiver buffer common-mode voltage.	VTT_OP80V VTT_OP75V VTT_OP70V VTT_OP65V VTT_OP60V VTT_OP55V VTT_OP50V VTT_OP35V VTT_PUP_WEAK VTT_PDN_WEAK TRISTATE1 VTT_PDN_STRONG VTT_PUP_STRONG TRISTATE2 TRISTATE3 TRISTATE4	Pin
XCVR_RX_ENABLE_LINEAR_EQUALIZER_PCIEMODE	Receiver Linear Equalizer Control (PCI Express)	If enabled equalizer gain control is driven by the PCS block for PCI Express. If disabled equalizer gain control is determined by the XCVR_RX_LINEAR_EQUALIZER_SETTING assignment.	TRUE FALSE	Pin
XCVR_RX_EQ_BW_SEL	Receiver Equalizer Gain Bandwidth Select	Sets the gain peaking frequency for the equalizer. For data-rates of less than 6.5Gbps set to HALF. For higher data-rates set to FULL.	FULL HALF	Pin
XCVR_RX_SD_ENABLE	Receiver Signal Detection Unit Enable/Disable	Enables or disables the receiver signal detection unit.	TRUE FALSE	Pin
XCVR_RX_SD_OFF	Receiver Cycle Count Before Signal Detect Block Declares Loss Of Signal	Number of parallel cycles to wait before the signal detect block declares loss of signal.	0 –29	Pin
XCVR_RX_SD_ON	Receiver Cycle Count Before Signal Detect Block Declares Presence Of Signal	Number of parallel cycles to wait before the signal detect block declares presence of signal.	0 –16	Pin
XCVR_RX_SD_THRESHOLD	Receiver Signal Detection Voltage Threshold	Specifies signal detection voltage threshold level.	0 –7	Pin

Table 3–8. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 3)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_TX_COMMON_MODE_VOLTAGE	Transmitter Common Mode Driver Voltage	Transmitter common-mode driver voltage	VOLT_OP80V VOLT_OP75V VOLT_OP70V VOLT_OP65V VOLT_OP60V VOLT_OP55V VOLT_OP50V VOLT_OP35V PULL_UP PULL_DOWN TRISTATED1 GROUNDED PULL_UP_TO_VCCELA TRISTATED2 TRISTATED3 TRISTATED4	Pin
XCVR_TX_PRE_EMP_1ST_POST_TAP	Transmitter Preemphasis First Post-Tap	Specifies the first post-tap setting value.	0 –31	Pin
XCVR_TX_PRE_EMP_2ND_POST_TAP	Transmitter Preemphasis Second Post-Tap	Specifies the second post-tap setting value.	0 –15	Pin
XCVR_TX_PRE_EMP_INV_2ND_TAP	Transmitter Preemphasis Second Tap Invert	Inverts the transmitter pre-emphasis 2nd post tap.	TRUE FALSE	Pin
XCVR_TX_PRE_EMP_INV_PRE_TAP	Transmitter Preemphasis Pre Tap Invert	Inverts the transmitter pre-emphasis pre-tap.	TRUE FALSE	Pin
XCVR_TX_PRE_EMP_PRE_TAP	Transmitter Preemphasis Pre-Tap	Specifies the pre-tap pre-emphasis setting.	0 –15	Pin
XCVR_TX_RX_DET_ENABLE	Transmitter's Receiver Detect Block Enable	Enables or disables the receiver detector circuit at the transmitter.	TRUE FALSE	Pin
XCVR_TX_RX_DET_MODE	Transmitter's Receiver Detect Block Mode	Sets the mode for receiver detect block	0 –15	Pin
XCVR_TX_RX_DET_OUTPUT_SEL	Transmitter's Receiver Detect Block QPI/PCI Express Control	Determines QPI or PCI Express mode for the Receiver Detect block.	RX_DET_QPI_OUT RX_DET_PCIE_OUT	Pin
XCVR_TX_VOD	Transmitter Differential Output Voltage	Differential output voltage setting. The values are monotonically increasing with the driver main tap current strength.	0–63 50	Pin
XCVR_TX_VOD_PRE_EMP_CTRL_SRC	Transmitter V _{OD} /Preemphasis Control Source	When set to DYNAMIC_CTL, the PCS block controls the V _{OD} and preemphasis coefficients for PCI Express. When this assignment is set to RAM_CTL the V _{OD} and preemphasis are controlled by other assignments, such as XCVR_TX_PRE_EMP_1ST_POST_TAP.	DYNAMIC_CTL RAM_CTL	Pin

For more information about the Pin Planner, refer to [About the Pin Planner](#) in Quartus II Help. For more information about the Assignment Editor, refer to [About the Assignment Editor](#) in Quartus II Help.

For more information about Quartus II Settings, refer to [Quartus II Settings File Manual](#).

Interfaces

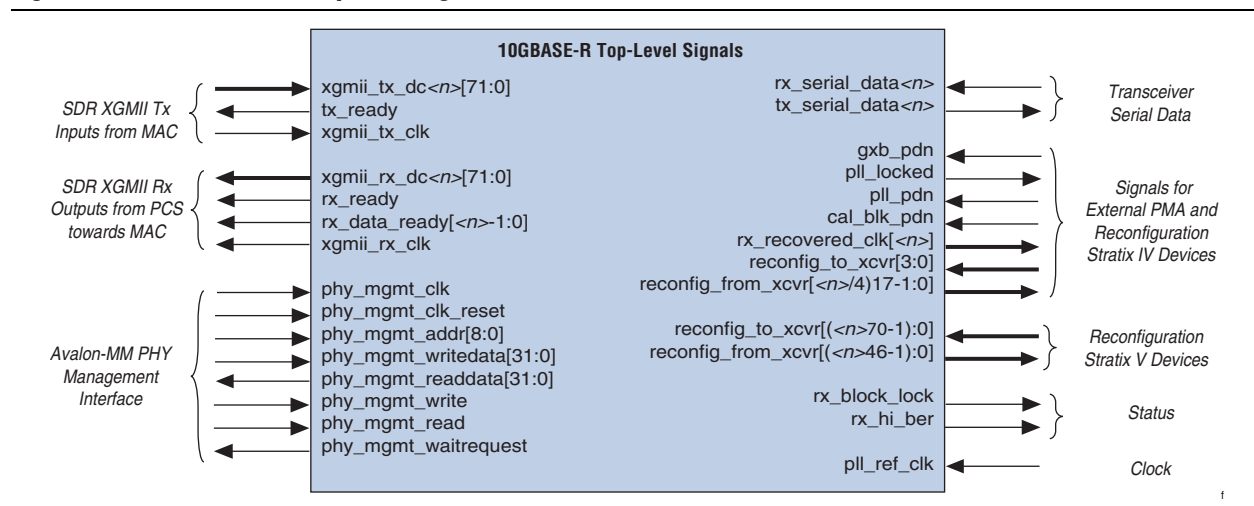
This section describes interfaces of the 10GBASE-R PHY IP Core. It includes the following topics:

- Ports
- Register Interface
- Dynamic Reconfiguration

Ports

Figure 3-3 illustrates the top-level signals of the 10BASE-R PHY.

Figure 3-3. 10GBASE-R PHY Top-Level Signals



The **block diagram** shown in the GUI labels the external pins with the *interface type* and places the *interface name* inside the box. The interface type and name are used in the Hardware Component Description File (`_hw.tcl`). If you turn on **Show signals**, the **block diagram** displays all top-level signal names.

For more information about `_hw.tcl` files refer to the [Component Interface Tcl Reference](#) chapter in volume 1 of the *Quartus II Handbook*.

The following sections describe the signals in each interface.

SDR XGMII TX Interface

Table 3–9 describes the signals in the SDR XGMII TX interface. These signals are driven from the MAC to the PCS. This is an Avalon-ST sink interface.

Table 3–9. SDR XGMII TX Inputs ⁽¹⁾

Signal Name	Direction	Description
xgmii_tx_dc[<n>71:0]	Sink	Contains 8 lanes of data and control for XGMII. Each lane consists of 8 bits of data and 1 bit of control. ■ Lane 0–[7:0]/[8] ■ Lane 1–[16:9]/[17] ■ Lane 2–[25:18]/[26] ■ Lane 3–[34:27]/[35] ■ Lane 4–[43:36]/[44] ■ Lane 5–[52:45]/[53] ■ Lane 6–[61:54]/[62] ■ Lane 7–[70:63]/[71] Refer to Table 3–10 for the mapping of the xgmii_tx_dc data and control to the xgmii_sdr_data and xgmii_sdr_ctrl signals.
tx_ready	Output	Asserted when the TX channel is ready to transmit data. Because the readyLatency on this Avalon-ST interface is 0, the MAC may drive tx_ready as soon as it comes out of reset.
xgmii_tx_clk	Input	The XGMII TX clock which runs at 156.25 MHz.

Note to Table 3–9:

(1) <n> is the channel number



For more information about the Avalon-ST protocol, including timing diagrams, refer to the *Avalon Interface Specifications*.

Table 3–10 provides the mapping from the XGMII TX interface to the XGMII SDR interface.

Table 3–10. Mapping from XGMII TX Bus to XGMII SDR Bus (Part 1 of 2)

Signal Name	XGMII Signal Name	Description
xgmii_tx_dc[7:0]	xgmii_sdr_data[7:0]	Lane 0 data
xgmii_tx_dc[8]	xgmii_sdr_ctrl[0]	Lane 0 control
xgmii_tx_dc[16:9]	xgmii_sdr_data[15:8]	Lane 1 data
xgmii_tx_dc[17]	xgmii_sdr_ctrl[1]	Lane 1 control
xgmii_tx_dc[25:18]	xgmii_sdr_data[23:16]	Lane 2 data
xgmii_tx_dc[26]	xgmii_sdr_ctrl[2]	Lane 2 control
xgmii_tx_dc[34:27]	xgmii_sdr_data[31:24]	Lane 3 data
xgmii_tx_dc[35]	xgmii_sdr_ctrl[3]	Lane 3 control
xgmii_tx_dc[43:36]	xgmii_sdr_data[39:32]	Lane 4 data
xgmii_tx_dc[44]	xgmii_sdr_ctrl[4]	Lane 4 control
xgmii_tx_dc[52:45]	xgmii_sdr_data[47:40]	Lane 5 data

Table 3-10. Mapping from XGMII TX Bus to XGMII SDR Bus (Part 2 of 2)

Signal Name	XGMII Signal Name	Description
xgmii_tx_dc[53]	xgmii_sdr_ctrl[5]	Lane 5 control
xgmii_tx_dc[61:54]	xgmii_sdr_data[55:48]	Lane 6 data
xgmii_tx_dc[62]	xgmii_sdr_ctrl[6]	Lane 6 control
xgmii_tx_dc[70:63]	xgmii_sdr_data[63:56]	Lane 7 data
xgmii_tx_dc[71]	xgmii_sdr_ctrl[7]	Lane 7 control

SDR XGMII RX Interface

Table 3-11 describes the signals in the SDR XGMII RX interface. This is an Avalon-ST source interface. These signals are driven from the PCS to the MAC.

Table 3-11. SDR XGMII RX Inputs ⁽¹⁾

Signal Name	Direction	Description
xgmii_rx_dc<n>[71:0]	Source	Contains 8 lanes of data and control for XGMII. Each lane consists of 8 bits of data and 1 bit of control. ■ Lane 0-[7:0]/[8] ■ Lane 1-[16:9]/[17] ■ Lane 2-[25:18]/[26] ■ Lane 3-[34:27]/[35] ■ Lane 4-[43:36]/[44] ■ Lane 5-[52:45]/[53] ■ Lane 6-[61:54]/[62] ■ Lane 7-[70:63]/[71] Refer to Table 3-12 for the mapping of the xgmii_rx_dc data and control to the xgmii_sdr_data and xgmii_sdr_ctrl signals.
rx_ready	Output	Asserted when the RX reset is complete.
rx_data_ready [<n>-1:0]	Output	When asserted, indicates that the PCS is sending data to the MAC. Because the readyLatency on this Avalon-ST interface is 0, the MAC must be ready to receive data whenever this signal is asserted. After rx_ready is asserted indicating the exit from the reset state, the MAC should store xgmii_rx_dc<n>[71:0] in each cycle where rx_data_ready<n> is asserted.
xgmii_rx_clk	Output	This clock is generated by the same reference clock that is used to generate the transceiver clock. Its frequency is 156.25 MHz. Use this clock for the MAC interface to minimize the size of the FIFO between the MAC and SDR XGMII RX interface.

Note to Table 3-11:

(1) <n> is the channel number

Table 3–12 provides the mapping from the XGMII RX interface to the XGMII SDR interface.

Table 3–12. Mapping from XGMII RX Bus to the XGMII SDR Bus

Signal Name	XGMII Signal Name	Description
xgmii_rx_dc[7:0]	xgmii_sdr_data[7:0]	Lane 0 data
xgmii_rx_dc[8]	xgmii_sdr_ctrl[0]	Lane 0 control
xgmii_rx_dc[16:9]	xgmii_sdr_data[15:8]	Lane 1 data
xgmii_rx_dc[17]	xgmii_sdr_ctrl[1]	Lane 1 control
xgmii_rx_dc[25:18]	xgmii_sdr_data[23:16]	Lane 2 data
xgmii_rx_dc[26]	xgmii_sdr_ctrl[2]	Lane 2 control
xgmii_rx_dc[34:27]	xgmii_sdr_data[31:24]	Lane 3 data
xgmii_rx_dc[35]	xgmii_sdr_ctrl[3]	Lane 3 control
xgmii_rx_dc[43:36]	xgmii_sdr_data[39:32]	Lane 4 data
xgmii_rx_dc[44]	xgmii_sdr_ctrl[4]	Lane 4 control
xgmii_rx_dc[52:45]	xgmii_sdr_data[47:40]	Lane 5 data
xgmii_rx_dc[53]	xgmii_sdr_ctrl[5]	Lane 5 control
xgmii_rx_dc[61:54]	xgmii_sdr_data[55:48]	Lane 6 data
xgmii_rx_dc[62]	xgmii_sdr_ctrl[6]	Lane 6 control
xgmii_rx_dc[70:63]	xgmii_sdr_data[63:56]	Lane 7 data
xgmii_rx_dc[71]	xgmii_sdr_ctrl[7]	Lane 7 control

Status Interface

Table 3–13 describes signals that provide status information.

Table 3–13. 10GBASE-R Receive Status Outputs

Signal Name	Direction	Description
block_lock	Output	Asserted to indicate that the block synchronizer has established synchronization.
hi_ber	Output	Asserted by the BER monitor block to indicate a Sync Header high bit error rate greater than 10^{-4} .

Clocks, Reset, and Powerdown

The `phy_mgmt_clk_reset` signal is the global reset that resets the entire PHY. A positive edge on this signal triggers a reset.



Refer to the *Reset Control and Power Down* chapter in volume 2 of the *Stratix IV Device Handbook* for additional information about reset sequences in Stratix IV devices.

Figure 3–4. Stratix IV GT Clock Generation and Distribution

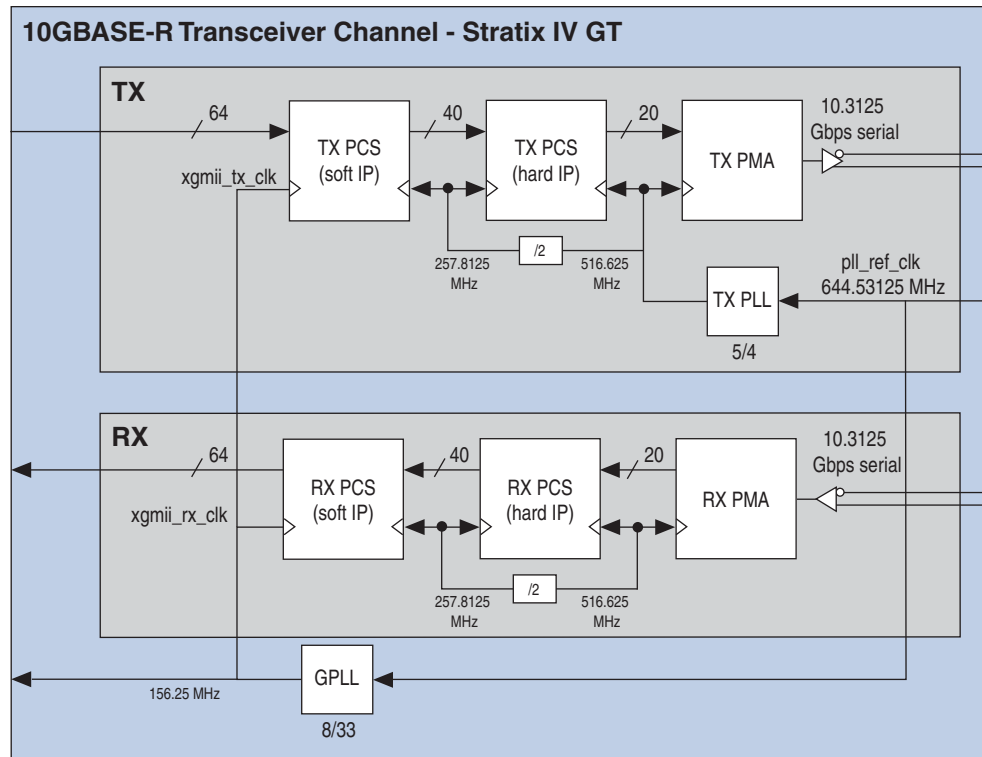
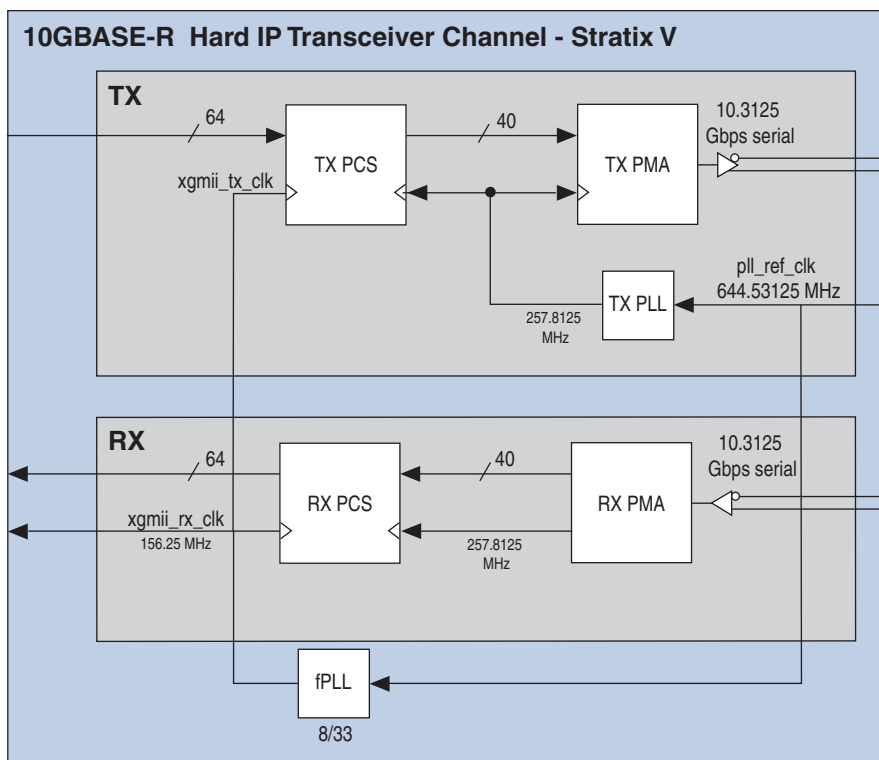


Figure 3–5 illustrates the clock generation and distribution for Stratix V devices.

Figure 3–5. Stratix V Clock Generation and Distribution



 To ensure proper functioning of the PCS, the maximum PPM difference between the `pll_ref_clk` and `xgmii_tx_clk` clock inputs is 100 PPM. You should use `xgmii_rx_clk` to drive `xgmii_tx_clk`. The CDR logic recovers 257.8125 MHz clock from the incoming data.

Table 3–15 describes the clock inputs.

Table 3–14. Clock Signals

Signal Name	Direction	Description
<code>pll_ref_clk</code>	Input	For Stratix IV GT devices, the TX PLL reference clock which must be 644.53125 MHz. For Stratix V devices, the TX PLL reference clock can be either 644.53125 MHz or 322.265625 MHz.

Serial Interface

Table 3-15 describes the input and outputs of the transceiver.

Table 3-15. Transceiver Serial Interface ⁽¹⁾

Signal Name	Direction	Description
rx_serial_data<n>	Input	Differential high speed serial input data using the PCML I/O standard. The clock is recovered from the serial data stream.
tx_serial_data<n>	Output	Differential high speed serial input data using the PCML I/O standard. The clock is embedded from the serial data stream.

Note to Table 3-15:

(1) <n> is the channel number.

Register Interface

The Avalon-MM PHY management interface provides access to the 10GBASER-R PHY PCS and PMA registers. You can use an embedded controller acting as an Avalon-MM master to send read and write commands to this Avalon-MM slave interface.

Table 3-16 describes the signals that comprise the Avalon-MM PHY Management interface.

Table 3-16. Avalon-MM PHY Management Interface

Signal Name	Direction	Description
phy_mgmt_clk	Input	The clock signal that controls the Avalon-MM PHY management, interface. For Stratix IV devices, the frequency range is 37.5–50 MHz. There is no frequency restriction for Stratix V devices; however, if you plan to use the same clock for the PHY management interface and transceiver reconfiguration, you must restrict the frequency range of phy_mgmt_clk to 100–125 MHz to meet the specification for the transceiver reconfiguration clock.
phy_mgmt_clk_reset	Input	Global reset signal that resets the entire 10GBASE-R PHY. This signal is active high and level sensitive.
phy_mgmt_addr[8:0]	Input	9-bit Avalon-MM address. Refer to for the address fields.
phy_mgmt_writedata[31:0]	Input	Input data.
phy_mgmt_readdata[31:0]	Output	Output data.
phy_mgmt_write	Input	Write signal. Asserted high.
phy_mgmt_read	Input	Read signal. Asserted high.
phy_mgmt_waitrequest	Output	When asserted, indicates that the Avalon-MM slave interface is unable to respond to a read or write request. When asserted, control signals to the Avalon-MM slave interface must remain constant.



Refer to the “Typical Slave Read and Write Transfers” and “Master Transfers” sections in the “Avalon Memory-Mapped Interfaces” chapter of the *Avalon Interface Specifications* for timing diagrams.

Register Descriptions

Table 3–17 specifies the registers that you can access over the Avalon-MM PHY management interface using word addresses and a 32-bit embedded processor. A single address space provides access to all registers.



Writing to reserved or undefined register addresses may have undefined side effects.

Table 3–17. 10GBASE-R Register Descriptions (Part 1 of 3)

Word Addr	Bit	R/W	Name	Description
PMA Common Control and Status - working in 12.0.				
0x021	[31:0]	RW	cal_blk_powerdown	Writing a 1 to channel <n> powers down the calibration block for channel <n>. This register is only available if you select Use external PMA control and reconfig on the Additional Options tab of the GUI.
0x022	[31:0]	RO	pma_tx_pll_is_locked	Bit[P] indicates that the TX clock multiplier unit CMU PLL [P] is locked to the input reference clock. This register is only available if you select Use external PMA control and reconfig on the Additional Options tab of the GUI.
Reset Control Registers—Automatic Reset Controller				
0x041	[31:0]	RW	reset_ch_bitmask	Reset controller channel bitmask for digital resets. The default value is all 1 s. Channel <n> can be reset when bit<n> = 1. Channel <n> cannot be reset when bit<n>=0.
0x042	[1:0]	WO	reset_control (write)	Writing a 1 to bit 0 initiates a TX digital reset using the reset controller module. The reset affects channels enabled in the reset_ch_bitmask. Writing a 1 to bit 1 initiates a RX digital reset of channels enabled in the reset_ch_bitmask. Both bits 0 and 1 self-clear.
		RO	reset_status (read)	Reading bit 0 returns the status of the reset controller TX ready bit. Reading bit 1 returns the status of the reset controller RX ready bit.

Table 3-17. 10GBASE-R Register Descriptions (Part 2 of 3)

Word Addr	Bit	R/W	Name	Description
0x044	[31:0]	RW	reset_fine_control	You can use the <code>reset_fine_control</code> register to create your own reset sequence. The reset control module, illustrated in Figure 1-1 on page 1-2 , performs a standard reset sequence at power on and whenever the <code>phy_mgmt_clk_reset</code> is asserted. Bits [31:4,0] are reserved.
	[31:4,0]	RW	Reserved	It is safe to write 0s to reserved bits.
	[1]	RW	reset_tx_digital	Writing a 1 causes the internal TX digital reset signal to be asserted, resetting all channels enabled in <code>reset_ch_bitmask</code> . You must write a 0 to clear the reset condition.
	[2]	RW	reset_rx_analog	Writing a 1 causes the internal RX digital reset signal to be asserted, resetting the RX analog logic of all channels enabled in <code>reset_ch_bitmask</code> . You must write a 0 to clear the reset condition.
	[3]	RW	reset_rx_digital	Writing a 1 causes the RX digital reset signal to be asserted, resetting the RX digital channels enabled in <code>reset_ch_bitmask</code> . You must write a 0 to clear the reset condition.
PMA Channel Control and Status				
0x061	[31:0]	RW	phy_serial_loopback	Writing a 1 to channel <i><n></i> puts channel <i><n></i> in serial loopback mode. For information about pre- or post-CDR serial loopback modes, refer to “Loopback Modes” on page 10-39 .
0x064	[31:0]	RW	pma_rx_set_locktodata	When set, programs the RX CDR PLL to lock to the incoming data. Bit <i><n></i> corresponds to channel <i><n></i> .
0x065	[31:0]	RW	pma_rx_set_locktoref	When set, programs the RX CDR PLL to lock to the reference clock. Bit <i><n></i> corresponds to channel <i><n></i> .
0x066	[31:0]	R	pma_rx_is_lockedtodata	When asserted, indicates that the RX CDR PLL is locked to the RX data, and that the RX CDR has changed from LTR to LTD mode. Bit <i><n></i> corresponds to channel <i><n></i> .
0x067	[31:0]	R	pma_rx_is_lockedtoref	When asserted, indicates that the RX CDR PLL is locked to the reference clock. Bit <i><n></i> corresponds to channel <i><n></i> .
10GBASE-R PCS				
0x080	[31:0]	RW	INDIRECT_ADDR	Provides for indirect addressing of all PCS control and status registers. Use this register to specify the logical channel number of the PCS channel you want to access.
0x081	[2]	RW	RCLR_ERRBLK_CNT	When set to 1, clears the error block count register. To block: Block synchronizer
	[3]	RW	RCLR_BER_COUNT	When set to 1, clears the bit error rate (BER) register. This bit is only for Stratix IV devices. To block: BER monitor

Table 3–17. 10GBASE-R Register Descriptions (Part 3 of 3)

Word Addr	Bit	R/W	Name	Description
0x082	[0]	R	PCS_STATUS	For Stratix IV devices, when asserted indicates that the PCS link is up. This bit is only available for Stratix IV devices.
	[1]	R	HI_BER	When asserted by the BER monitor block, indicates that the PCS is recording a high BER. This bit is only for Stratix IV devices. From block: BER monitor
	[2]	R	BLOCK_LOCK	When asserted by the block synchronizer, indicates that the PCS is locked to received blocks. From Block: Block synchronizer
	[3]	R	TX_FIFO_FULL	When asserted, indicates the TX FIFO is full. From block: TX FIFO
	[4]	R	RX_FIFO_FULL	When asserted, indicates the RX FIFO is full. From block: RX FIFO
	[5]	R	RX_SYNC_HEAD_ERROR	For Stratix V devices, when asserted, indicates an RX synchronization error. This signal is Stratix V devices only.
	[6]	R	RX_SCRAMBLER_ERROR	For Stratix V devices, when asserted, indicates an RX scrambler error. This signal is Stratix V devices only.
	[7]	R	RX_DATA_READY	When asserted indicates that the RX interface is ready to send out received data. From block: 10 Gbps Receiver PCS
0x083	[5:0]	R	BER_COUNT[5:0]	For Stratix IV devices only, records the bit error rate (BER). From block: BER monitor
	[13:6]	R	ERROR_BLOCK_COUNT[7:0]	For Stratix IV devices only, records the number of blocks that contain errors. From Block: Block synchronizer

Dynamic Reconfiguration

This section includes information about dynamic reconfiguration of the transceivers in Stratix IV and Stratix V devices.

Dynamic Reconfiguration for Stratix IV Devices

Table 3–18 describes the additional top-level signals 10GBASE-R PHY IP core when the configuration includes external modules for PMA control and reconfiguration. You enable this configuration by turning on **Use external PMA control and reconfig** available for Stratix IV GT devices. This configuration is illustrated in Figure 3–2 on page 3–2.

Table 3–18. External PMA and Reconfiguration Signals (Part 1 of 2)

Signal Name	Direction	Description
gxb_pdn	Input	When asserted, powers down the entire GX block. Active high.
pll_locked	Output	When asserted, indicates that the PLL is locked. Active high.
pll_pdn	Input	When asserted, powers down the TX PLL. Active high.

Table 3–18. External PMA and Reconfiguration Signals (Part 2 of 2)

Signal Name	Direction	Description
cal_blk_pdn	Input	When asserted, powers down the calibration block. Active high.
cal_blk_clk	Input	Calibration clock. For Stratix IV devices only. It must be in the range 37.5–50 MHz. You can use the same clock for the phy_mgmt_clk and the cal_blk_clk.
rx_recovered_clk[<n>:0]	Output	This is the RX clock, which is recovered from the received data stream.
reconfig_to_xcvr[3:0]	Input	Reconfiguration signals from the Transceiver Reconfiguration Controller to the PHY device. This signal is only available in Stratix IV devices.
reconfig_from_xcvr [(<n>/4)17-1:0]	Output	Reconfiguration RAM. The PHY device drives this RAM data to the transceiver reconfiguration IP. This signal is only available in Stratix IV devices.

Table 3–19 describes the signals in the reconfiguration interface. This interface uses the Avalon-MM PHY Management interface clock.

Table 3–19. Reconfiguration Interface

Signal Name	Direction	Description
reconfig_to_xcvr [(<n>70-1):0]	Sink	Reconfiguration signals from the Transceiver Reconfiguration Controller. <n> grows linearly with the number of reconfiguration interfaces. This signal is only available in Stratix V devices.
reconfig_from_xcvr [(<n>46-1):0]	Source	Reconfiguration signals to the Transceiver Reconfiguration Controller. <n> grows linearly with the number of reconfiguration interfaces. This signal is only available in Stratix V devices.

Dynamic Reconfiguration for Stratix V Devices

As silicon progresses towards smaller process nodes, circuit performance is affected more by variations due to process, voltage, and temperature (PVT). These process variations result in analog voltages that can be offset from required ranges. The calibration performed by the dynamic reconfiguration interface compensates for variations due to PVT.

For Stratix V devices, each channel and each TX PLL have separate dynamic reconfiguration interfaces. The parameter editor provides informational messages on the connectivity of these interfaces. Example 3–1 shows the messages for a single duplex channel.

Although you must initially create a separate reconfiguration interface for each channel and TX PLL in your design, when the Quartus II software compiles your design, it reduces the number of reconfiguration interfaces by merging reconfiguration interfaces. The synthesized design typically includes a reconfiguration interface for three channels. Allowing the Quartus II software to merge reconfiguration interfaces gives the Fitter more flexibility in placing transceiver channels. For more information about transceiver reconfiguration refer to [Chapter 10, Transceiver Reconfiguration Controller](#).

Example 3-1. Informational Messages for the Transceiver Reconfiguration Interface

PHY IP will require 2 reconfiguration interfaces for connection to the external reconfiguration controller.
Reconfiguration interface offset 0 is connected to the transceiver channel.
Reconfiguration interface offset 1 is connected to the transmit PLL.

TimeQuest Timing Constraints

The timing constraints for Stratix IV GX designs are in `alt_10gbaser_phy.sdc`. If your design does not meet timing with these constraints, use LogicLock™ for the `alt_10gbaser_pcs` block. You can also apply LogicLock to the `alt_10gbaser_pcs` and slightly expand the lock region to meet timing.

 For more information about LogicLock, refer to [About LogicLock Regions](#) in Quartus II Help.

 For Stratix V devices, timing constraints are built into the HDL code.

[Example 3-2](#) provides the Synopsys Design Constraints File (`.sdc`) timing constraints for the 10GBASE-R IP core. To pass timing analysis, you must decouple the clocks in different time domains. Be sure to verify the each clock domain is correctly buffered in the top level of your design. You can find the `.sdc` file in your top-level working directory. This is the same directory that includes your top-level `.v` or `.vhd` file.

Example 3-2. Synopsys Design Constraints for Clocks

```
*****
# Timing Information
*****
set_time_format -unit ns -decimal_places 3
*****
# Create Clocks
*****
create_clock -name {xgmii_tx_clk} -period 6.400 -waveform { 0.000 3.200 } [get_ports
{xgmii_tx_clk}]
create_clock -name {phy_mgmt_clk} -period 20.00 -waveform { 0.000 10.000 } [get_ports
{phy_mgmt_clk}]
create_clock -name {pll_ref_clk} -period 1.552 -waveform { 0.000 0.776 } [get_ports
{ref_clk}]
#derive_pll_clocks
derive_pll_clocks -create_base_clocks
#derive_clocks -period "1.0"
```

Synopsys Design Constraints for Clocks (Continued)

```

*****
# Create Generated Clocks
*****
create_generated_clock -name pll_mac_clk -source [get_pins -compatibility_mode
{*altpll_component|auto_generated|pll1|clk[0]}]
create_generated_clock -name pma_tx_clk -source [get_pins -compatibility_mode
{*siv_alt_pma|pma_direct|auto_generated|transmit_pcs0|clkout}]
*****
## Set Clock Latency
*****
# Set Clock Uncertainty
*****
derive_clock_uncertainty
set_clock_uncertainty -from [get_clocks
{*siv_alt_pma|pma_ch*.pma_direct|receive_pcs*|clkout}] -to pll_ref_clk -setup 0.1
set_clock_uncertainty -from [get_clocks
{*siv_alt_pma|pma_direct|auto_generated|transmit_pcs0|clkout}] -to pll_ref_clk -setup
0.08
set_clock_uncertainty -from [get_clocks
{*siv_alt_pma|pma_ch*.pma_direct|receive_pcs*|clkout}] -to pll_ref_clk -hold 0.1
set_clock_uncertainty -from [get_clocks
{*siv_alt_pma|pma_direct|auto_generated|transmit_pcs0|clkout}] -to pll_ref_clk -hold
0.08
*****
# Set Input Delay
*****
# Set Output Delay
*****
# Set Clock Groups
*****
set_clock_groups -exclusive -group phy_mgmt_clk -group xgmii_tx_clk -group [get_clocks
{*siv_alt_pma|pma_ch*.pma_direct|transmit_pcs*|clkout}] -group [get_clocks
{*siv_alt_pma|pma_ch*.pma_direct|receive_pcs*|clkout}] -group [get_clocks
{*pll_siv_xgmii_clk|altpll_component|auto_generated|pll1|clk[0]}]
## *****
# Set False Path
*****
set_false_path -from {*siv_10gbaser_xcvr*clk_reset_ctrl|rx_pma_rstn} -to [get_clocks
{{*siv_alt_pma|pma_ch*.pma_direct|transmit_pcs*|clkout}
{*siv_alt_pma|pma_ch*.pma_direct|receive_pcs*|clkout}
{*pll_siv_xgmii_clk|altpll_component|auto_generated|pll1|clk[0]} phy_mgmt_clk
xgmii_tx_clk}]
set_false_path -from {*siv_10gbaser_xcvr*clk_reset_ctrl|rx_usr_rstn} -to [get_clocks
{{*siv_alt_pma|pma_ch*.pma_direct|transmit_pcs*|clkout}
{*siv_alt_pma|pma_ch*.pma_direct|transmit_pcs*|clkout}
{*pll_siv_xgmii_clk|altpll_component|auto_generated|pll1|clk[0]} phy_mgmt_clk
xgmii_tx_clk}]
#

```

Synopsys Design Constraints for Clocks (Continued)

```

set_false_path -from {*siv_10gbaser_xcvr*clk_reset_ctrl|tx_pma_rstn} -to [get_clocks
{{*siv_alt_pma|pma_ch*.pma_direct|receive_pcs*|clkout}}
{{*siv_alt_pma|pma_ch*.pma_direct|transmit_pcs*|clkout}}
{{*pll_siv_xgmii_clk|altpll_component|auto_generated|pll1|clk[0]} phy_mgmt_clk
xgmii_tx_clk}}]
set_false_path -from {*siv_10gbaser_xcvr*clk_reset_ctrl|tx_usr_rstn} -to [get_clocks
{{*siv_alt_pma|pma_ch*.pma_direct|receive_pcs*|clkout}}
{{*siv_alt_pma|pma_ch*.pma_direct|transmit_pcs*|clkout}}
{{*pll_siv_xgmii_clk|altpll_component|auto_generated|pll1|clk[0]} phy_mgmt_clk
xgmii_tx_clk}}]
set_false_path -from {*siv_10gbaser_xcvr*rx_analog_rst_lego|rinit} -to [get_clocks
{{*siv_alt_pma|pma_ch*.pma_direct|receive_pcs*|clkout}}
{{*siv_alt_pma|pma_ch*.pma_direct|transmit_pcs*|clkout}}
{{*pll_siv_xgmii_clk|altpll_component|auto_generated|pll1|clk[0]} phy_mgmt_clk
xgmii_tx_clk}}]
set_false_path -from {*siv_10gbaser_xcvr*rx_digital_rst_lego|rinit} -to [get_clocks
{{*siv_alt_pma|pma_ch*.pma_direct|receive_pcs*|clkout}}
{{*siv_alt_pma|pma_ch*.pma_direct|transmit_pcs*|clkout}}
{{*pll_siv_xgmii_clk|altpll_component|auto_generated|pll1|clk[0]} phy_mgmt_clk
xgmii_tx_clk}}]
#*****
# Set Multicycle Paths
#*****
#*****
# Set Maximum Delay
#*****
#*****
# Set Minimum Delay
#*****
#*****
# Set Input Transition
#*****

```



This **.sdc** file is only applicable to the 10GBASE-R PHY IP core when compiled in isolation. You can use it as a reference to help in creating your own **.sdc** file.

Simulation Files and Example Testbench

Refer to [“Running a Simulation Testbench” on page 1–4](#) for a description of the directories and files that the Quartus II software creates automatically when you generate your 10GBASE-R PHY IP core.

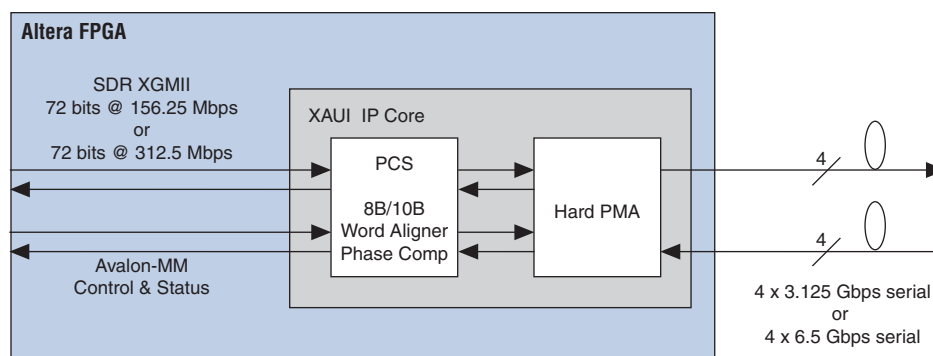


Refer to the [Altera wiki](#) for an example testbench that you can use as a starting point in creating your own verification environment.

The Altera XAUI PHY IP core implements the *IEEE 802.3 Clause 48* specification to extend the operational distance of the XGMII interface and reduce the number of interface signals. XAUI extends the physical separation possible between the 10 Gbps Ethernet MAC function implemented in an Altera FPGA and the Ethernet standard PHY component on a PCB to one meter. The XAUI IP core accepts 72-bit data (single data rate–SDR XGMII) from the application layer at either 156.25 Mbps or 312.5 Mbps. The serial interface runs at either 4×3.125 Gbps or 4×6.25 Gbps (DDR XAUI option).

Figure 4–1 illustrates the top-level blocks of the XAUI PHY IP core.

Figure 4–1. XAUI PHY IP Core



Note to Figure 4–1:

- (1) Refer to Table 4–2 on page 4–2 for a list of supported devices.

For Stratix IV GX and GT devices, you can choose a hard XAUI physical coding sublayer (PCS) and physical media attachment (PMA), or a soft XAUI PCS and PMA in low latency mode. You can also combine both hard and soft PCS configurations in the same device, using all channels in a transceiver bank. The PCS is only available in soft logic for Stratix V devices.



For more detailed information about the XAUI transceiver channel datapath, clocking, and channel placement, refer to the “XAUI” section in the *Transceiver Protocol Configurations in Stratix V Devices* chapter of the *Stratix V Device Handbook*.

Release Information

Table 4-1 provides information about this release of the XAUI PHY IP core.

Table 4-1. XAUI Release Information

Item	Description
Version	11.1
Release Date	November 2011
Ordering Codes ⁽¹⁾	IP-XAUIPCS (primary)–Soft PCS IPR-XAUIPCS (renewal)–Soft PCS
Product ID	00D7
Vendor ID	6AF7

Note to Table 4-1:

(1) No ordering codes or license files are required for the hard PCS and PMA PHY in Arria II GX, Cyclone® IV GX, or Stratix IV GX or GT devices.

Device Family Support

IP cores provide either final or preliminary support for target Altera device families. These terms have the following definitions:

- *Final support*—Verified with final timing models for this device.
- *Preliminary support*—Verified with preliminary timing models for this device.

Table 4-2 shows the level of support offered by the XAUI IP core for Altera device families.

Table 4-2. Device Family Support

Device Family	Support
XAUI	
Arria II GX –Hard PCS and PMA	Final
Arria II GZ–Hard PCS and PMA	Final
Cyclone IV GX–Hard PCS and PMA	Final
HardCopy® IV	Final
Stratix IV GX and GT devices–Soft or hard PCS and PMA	Final
Stratix V devices–Soft PCS + PMA	Preliminary
Other device families	No support
DXAUI	
Stratix IV GX and GT	Final
Other device families	No support

Performance and Resource Utilization

Table 4–3 shows the typical expected device resource utilization for different configurations using the current version of the Quartus II software targeting a Stratix IV GX (EP4SG230KF40C2ES) device.

Table 4–3. XAUI PHY Performance and Resource Utilization—Stratix IV GX Device

Implementation	Number of 3.125 Gbps Channels	Worst-Case Frequency	Combinational ALUTs	Dedicated Registers	Memory Bits
Soft XAUI	4	183.18 MHz	4500	3200	5100
Hard XAUI	4	400 MHz	2000	1300	0

Parameter Settings

To configure the XAUI IP core in the parameter editor, click **Installed Plug-Ins > Interfaces > Ethernet > XAUI PHY v11.1**.

This section describes the XAUI PHY IP core parameters, which you can set using the parameter editor. Table 4–4 lists the settings available on **General Options** tab.

Table 4–4. General Options (Part 1 of 2)

Name	Value	Description
Device family	Arria II GX Cyclone IV GX Stratix IV Stratix V	The target device family.
Starting channel number	0–124	The physical starting channel number in the Altera device for channel 0 of this XAUI PHY. In Arria II GX, Cyclone IV GX, HardCopy IV and Stratix IV devices, this starting channel number must be 0 or a multiple of 4. There are no numbering restrictions for Stratix V devices. Assignment of the starting channel number is required for serial transceiver dynamic reconfiguration.
XAUI interface type	Hard XAUI Soft XAUI DDR XAUI	The following 3 interface types are available: ■ Hard XAUI—Implements the PCS and PMA in hard logic. Available for Arria II, Cyclone IV, HardCopy IV, and Stratix IV devices. ■ Soft XAUI—Implements the PCS in soft logic and the PMA in hard logic. Available for HardCopy IV, Stratix IV, and Stratix V devices. ■ DDR XAUI—Implements the PCS in soft logic and the PMA in hard logic. Both the application and serial interfaces run at twice the frequency of the Soft XAUI options. Available for HardCopy IV Stratix IV devices. All interface types include 4 channels.
PLL type	CMU ATX	You can select either the CMU or ATX PLL. The CMU PLL has a larger frequency range than the ATX PLL. The ATX PLL is designed to improve jitter performance and achieves lower channel-to-channel skew; however, it supports a narrower range of data rates and reference clock frequencies. Another advantage of the ATX PLL is that it does not use a transceiver channel, while the CMU PLL does. This parameter is available for Stratix IV soft and DDR XAUI, and Stratix V devices.

Table 4–4. General Options (Part 2 of 2)

Name	Value	Description
Base data rate	1 × Lane rate 2 × Lane rate 4 × Lane rate	The base data rate is the frequency of the clock input to the PLL. Select a base data rate that minimizes the number of PLLs required to generate all the clock s required for data transmission. By selecting an appropriate base data rate , you can change data rates by changing the divider used by the clock generation block. This parameter is available for Stratix V devices.
Number of XAUI interfaces	1	Specifies the number of XAUI interfaces. Only 1 is available in the current release.

Analog Options

The following sections explain how to specify analog options for supported device families.

Arria II GX, Cyclone IV GX, HardCopy IV and Stratix IV Devices

Arria II GX, Cyclone IV GX, and Stratix IV devices, you specify analog options on the Analog Options tab. Table 4–5 describes these options.

Table 4–5. PMA Analog Options (Part 1 of 2)

Name	Value	Description
Transmitter termination resistance	OCT_85_OHMS OCT_100_OHMS OCT_120_OHMS OCT_150_OHMS	Indicates the value of the termination resistor for the transmitter.
Transmitter VOD control setting	0–7	Sets V_{OD} for the various TX buffers.
Pre-emphasis pre-tap setting	0–7	Sets the amount of pre-emphasis on the TX buffer. Available for Stratix IV
Invert the pre-emphasis pre-tap polarity setting	On Off	Determines whether or not the pre-emphasis control signal for the pre-tap is inverted. If you turn this option on, the pre-emphasis control signal is inverted. Available for HardCopy IV and Stratix IV devices.
Pre-emphasis first post-tap setting	0–15	Sets the amount of pre-emphasis for the 1st post-tap.
Pre-emphasis second post-tap setting	0–7	Sets the amount of pre-emphasis for the 2nd post-tap. Available for HardCopy IV and Stratix IV devices.
Invert the pre-emphasis second post-tap polarity	On Off	Determines whether or not the pre-emphasis control signal for the second post-tap is inverted. If you turn this option on, the pre-emphasis control signal is inverted. Available for HardCopy IV and Stratix IV devices.
Receiver common mode voltage	Tri-state 0.82V 1.1v	Specifies the RX common mode voltage.
Receiver termination resistance	OCT_85_OHMS OCT_100_OHMS OCT_120_OHMS OCT_150_OHMS	Indicates the value of the termination resistor for the receiver. Cyclone IV supports 100 and 150.

Table 4–5. PMA Analog Options (Part 2 of 2)

Name	Value	Description
Receiver DC gain	0–4	Sets the equalization DC gain using one of the following settings: ■ 0–0 dB ■ 1–3 dB ■ 2–6 dB ■ 3–9 dB ■ 4–12 dB
Receiver static equalizer setting:	0–15	This option sets the equalizer control settings. The equalizer uses a pass band filter. Specifying a low value passes low frequencies. Specifying a high value passes high frequencies. Available for HardCopy IV and Stratix IV devices.

Stratix V Devices

You specify the analog parameters for Stratix V devices using the Quartus II Assignment Editor, the Pin Planner, or through the Quartus II Settings File (.qsf). The default values for analog options fall into three categories:

- *Global*— These parameters have default values that are independent of other parameter settings.
- *Computed*—These parameters have an initial default value that is recomputed based on other parameter settings.
- *Proxy*—These parameters have default values that are place holders. The Quartus II software selects these initial default values based on your design; however, Altera recommends that you replace these defaults with values that match your electrical board specification.

Table 4–6 lists the analog parameters for Stratix V devices whose original values are place holders for the values that match your electrical board specification. In Table 4–6, the default value of an analog parameter is shown in **bold** type.

Table 4–6. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 2)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_IO_PIN_TERMINATION	Transceiver I/O Pin Termination	Specifies the intended on-chip termination value for the specified transceiver pin. Use External Resistor if you intend to use off-chip termination.	85_OHMS 100_OHMS 120_OHMS 150_OHMS EXTERNAL_RESISTOR	Pin
XCVR_REFCLK_PIN_TERMINATION	Transceiver Dedicated Refclk Pin Termination	Specifies the intended termination value for the specified refclk pin.	DC_COUPLING_INTERNAL_100_OHM DC_COUPLING_EXTERNAL_RESISTOR AC_COUPLING	Pin

Table 4–6. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 2)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_RX_BYPASS_EQ_STAGES_234	Receiver Equalizer Stage 2, 3, 4 Bypass	Bypass continuous time equalizer stages 2, 3, and 4 to save power. This setting eliminates significant AC gain on the equalizer and is appropriate for chip-to-chip short range communication on a PCB.	ALL_STAGES_ENABLED BYPASS_STAGES	Pin
XCVR_TX_SLEW_RATE_CTRL	Transmitter Slew Rate Control	Specifies the slew rate of the output signal. The valid values span from the slowest rate to fastest rate with 1 representing the slowest rate.	1–5	Pin
XCVR_VCCA_VOLTAGE	VCCA_GXB Voltage	Configure the VCCA_GXB voltage for a GXB I/O pin by specifying the intended VCCA_GXB voltage for a GXB I/O pin. If you do not make this assignment the compiler automatically sets the correct VCCA_GXB voltage depending on the configured data rate, as follows: ■ Data rate <= 6.5 Gbps: 2_5V ■ Data rate > 6.5 Gbps: 3_0V or 3_3V for Stratix V ES silicon	2_5V 3_0V	Pin
XCVR_VCCR_VCCT_VOLTAGE	VCCR_GXB VCCT_GXB Voltage	Configure the VCCR_GXB and VCCT_GXB voltage for an GXB I/O pin by specifying the intended supply voltages for a GXB I/O pin. If your design uses decision feedback equalization (DFE) or adaptive equalization (AEQ), you must set this parameter to 1.0V. Otherwise, if you do not make this assignment the compiler automatically sets the correct VCCR_GXB and VCCT_GXB voltage depending on the configured data rate as follows: ■ Data rate <= 6.5 Gbps: 0_85V ■ Data rate > 6.5 Gbps: 1_0V	0_85V 1_0V	Pin

Table 4-7 lists the analog parameters with *global* or *computed* default values. You may want to optimize some of these settings. In Table 4-7, the default value is shown in **bold** type. For computed analog parameters, the default value listed is for the initial setting, not the recomputed setting.

Table 4-7. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 3)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
Analog Parameters with Global Default Value				
CDR_BANDWIDTH_PRESET	CDR Bandwidth Preset	Specifies the CDR bandwidth preset setting.	Auto Low Medium High	PLL instance
PLL_BANDWIDTH_PRESET	PLL Bandwidth Preset	Specifies the PLL bandwidth preset setting	Auto Low Medium High	PLL instance
XCVR_RX_DC_GAIN	Receiver Buffer DC Gain Control	Controls the amount of a stage receive-buffer DC gain.	0 –4	Pin
XCVR_RX_LINEAR_EQUALIZER_CONTROL	Receiver Linear Equalizer Control	Static control for the continuous time equalizer in the receiver buffer. The equalizer has 16 distinct settings from 0–15 corresponding to the increasing AC gain.	1 –16	Pin
Analog Parameter with Computed Default Value				
XCVR_RX_COMMON_MODE_VOLTAGE	Receiver Buffer Common Mode Voltage	Receiver buffer common-mode voltage.	VTT_OP80V VTT_OP75V VTT_OP70V VTT_OP65V VTT_OP60V VTT_OP55V VTT_OP50V VTT_OP35V VTT_PUP _WEAK VTT_PDN WEAK TRISTATE1 VTT_PDN_ STRONG VTT_PUP_ STRONG TRISTATE2 TRISTATE3 TRISTATE4	Pin
XCVR_RX_ENABLE_LINEAR_EQUALIZER_PCIEMODE	Receiver Linear Equalizer Control (PCI Express)	If enabled equalizer gain control is driven by the PCS block for PCI Express. If disabled equalizer gain control is determined by the XCVR_RX_LINEAR_EQUALIZER_SETTING assignment.	TRUE FALSE	Pin

Table 4-7. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 3)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_RX_EQ_BW_SEL	Receiver Equalizer Gain Bandwidth Select	Sets the gain peaking frequency for the equalizer. For data-rates of less than 6.5Gbps set to HALF. For higher data-rates set to FULL.	FULL HALF	Pin
XCVR_RX_SD_ENABLE	Receiver Signal Detection Unit Enable/Disable	Enables or disables the receiver signal detection unit.	TRUE FALSE	Pin
XCVR_RX_SD_OFF	Receiver Cycle Count Before Signal Detect Block Declares Loss Of Signal	Number of parallel cycles to wait before the signal detect block declares loss of signal.	0–29	Pin
XCVR_RX_SD_ON	Receiver Cycle Count Before Signal Detect Block Declares Presence Of Signal	Number of parallel cycles to wait before the signal detect block declares presence of signal.	0–16	Pin
XCVR_RX_SD_THRESHOLD	Receiver Signal Detection Voltage Threshold	Specifies signal detection voltage threshold level.	0 –7	Pin
XCVR_TX_COMMON_MODE_VOLTAGE	Transmitter Common Mode Driver Voltage	Transmitter common-mode driver voltage	VOLT_OP80V VOLT_OP75V VOLT_OP70V VOLT_OP65V VOLT_OP60V VOLT_OP55V VOLT_OP50V VOLT_OP35V PULL_UP PULL_DOWN TRISTATED1 GROUNDED PULL_UP_TO_VCCELA TRISTATED2 TRISTATED3 TRISTATED4	Pin
XCVR_TX_PRE_EMP_1ST_POST_TAP	Transmitter Preemphasis First Post-Tap	Specifies the first post-tap setting value.	0 –31	Pin
XCVR_TX_PRE_EMP_2ND_POST_TAP	Transmitter Preemphasis Second Post-Tap	Specifies the second post-tap setting value.	0–15	Pin
XCVR_TX_PRE_EMP_INV_2ND_TAP	Transmitter Preemphasis Second Tap Invert	Inverts the transmitter pre-emphasis 2nd post tap.	TRUE FALSE	Pin
XCVR_TX_PRE_EMP_INV_PRE_TAP	Transmitter Preemphasis Pre Tap Invert	Inverts the transmitter pre-emphasis pre-tap.	TRUE FALSE	Pin
XCVR_TX_PRE_EMP_PRE_TAP	Transmitter Preemphasis Pre-Tap	Specifies the pre-tap pre-emphasis setting.	0 –15	Pin
XCVR_TX_RX_DET_ENABLE	Transmitter's Receiver Detect Block Enable	Enables or disables the receiver detector circuit at the transmitter.	TRUE FALSE	Pin
XCVR_TX_RX_DET_MODE	Transmitter's Receiver Detect Block Mode	Sets the mode for receiver detect block	0–15	Pin
XCVR_TX_RX_DET_OUTPUT_SEL	Transmitter's Receiver Detect Block QPI/PCI Express Control	Determines QPI or PCI Express mode for the Receiver Detect block.	RX_DET_QPI_OUT RX_DET_PCIE_OUT	Pin

Table 4-7. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 3)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_TX_VOD	Transmitter Differential Output Voltage	Differential output voltage setting. The values are monotonically increasing with the driver main tap current strength.	0–63 50	Pin
XCVR_TX_VOD_PRE_EMP_CTRL_SRC	Transmitter V _{OD} /Preemphasis Control Source	When set to DYNAMIC_CTL, the PCS block controls the V _{OD} and preemphasis coefficients for PCI Express. When this assignment is set to RAM_CTL the V _{OD} and preemphasis are controlled by other assignments, such as XCVR_TX_PRE_EMP_1ST_POST_TAP.	DYNAMIC_CTL RAM_CTL	Pin

❓ For more information about the Pin Planner, refer to [About the Pin Planner](#) in Quartus II Help. For more information about the Assignment Editor, refer to [About the Assignment Editor](#) in Quartus II Help.

🔧 For more information about Quartus II Settings, refer to [Quartus II Settings File Manual](#).

Advanced Options

Table 4-8 describes the settings available on the **Advanced Options** tab.

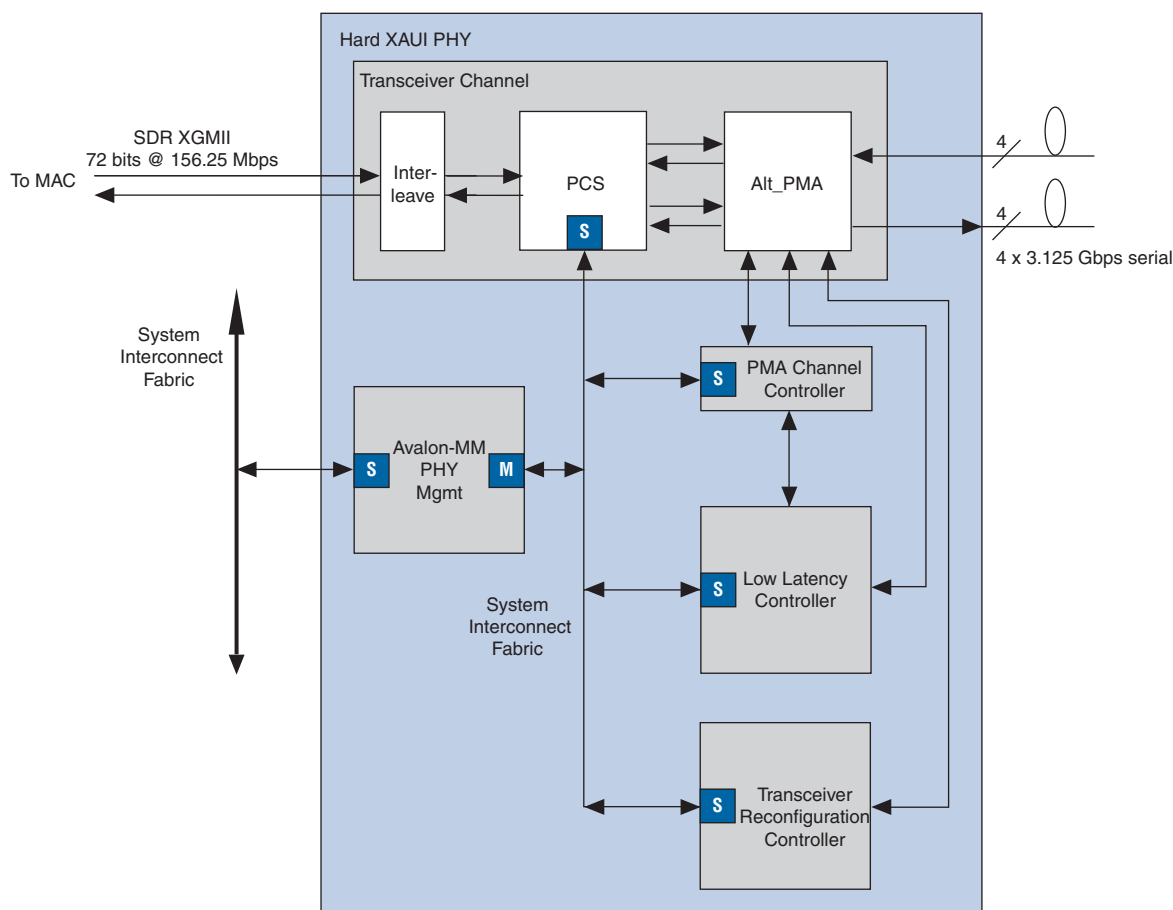
Table 4-8. Advanced Options

Name	Value	Description
Include control and status ports	On/Off	If you turn this option on, the top-level IP core include the status signals and digital resets shown in Figure 4-4 on page 4-12 and Figure 4-3 on page 4-11 . If you turn this option off, you can access control and status information using Avalon-MM interface to the control and status registers. The default setting is off.
External PMA control and configuration	On/Off	If you turn this option on, the PMA signals are brought up to the top level of the XAUI IP core. This option is useful if your design includes multiple instantiations of the XAUI PHY IP core. To save FPGA resources, you can instantiate the Low Latency PHY Controller and Transceiver Reconfiguration Controller IP cores separately in your design to avoid having these IP cores instantiated in each instance of the XAUI PHY IP core. If you turn this option off, the PMA signals remain internal to the core. The default setting is off. This option is available for Arria II GX, HardCopy IV and Stratix IV devices.
Enable rx_recovered_clk pin	On/Off	When you turn this option on, the RX recovered clock signal is an output signal.

Configurations

Figure 4-2 illustrates one configuration of the XAUI IP core. As this figure illustrates, if your variant includes a single instantiation of the XAUI IP core, the transceiver reconfiguration control logic is included in the XAUI PHY IP core. For Stratix V devices the Transceiver Reconfiguration Controller must always be external. Refer to [Chapter 10, Transceiver Reconfiguration Controller](#) for more information about this IP core.

Figure 4-2. XAUI PHY with Internal Transceiver Reconfiguration Control



Interfaces

This section describes interfaces of the XAUI PHY IP Core. It includes the following topics:

- Ports
- Registers
- Dynamic Reconfiguration

Ports

Figure 4-3 illustrates the top-level signals of the XAUI PHY IP core for the hard IP implementation. Figure 4-4 illustrates the top-level signals of the XAUI PHY IP core for the soft IP implementation. With the exception of the optional signals available for debugging and the signals for dynamic reconfiguration of the transceivers, the pinout of the two implementations is nearly identical. The DDR XAUI soft IP signals and behavior are the same as the soft IP implementation.



The **block diagram** shown in the GUI labels the external pins with the *interface type* and places the *interface name* inside the box. The interface type and name are used to define component interfaces in the `_hw.tcl`. If you turn on **Show signals**, the **block diagram** displays all top-level signal names.



For more information about `_hw.tcl` files refer to the [Component Interface Tcl Reference](#) chapter in volume 1 of the *Quartus II Handbook*.

Figure 4-3 illustrates the top-level signals of the XAUI PHY IP core for the hard IP implementation which is available for Arria II GX, Cyclone IV GX, HardCopy IV and Stratix IV GX devices.

Figure 4-3. XAUI Top-Level Signals—Hard IP PCS and PMA

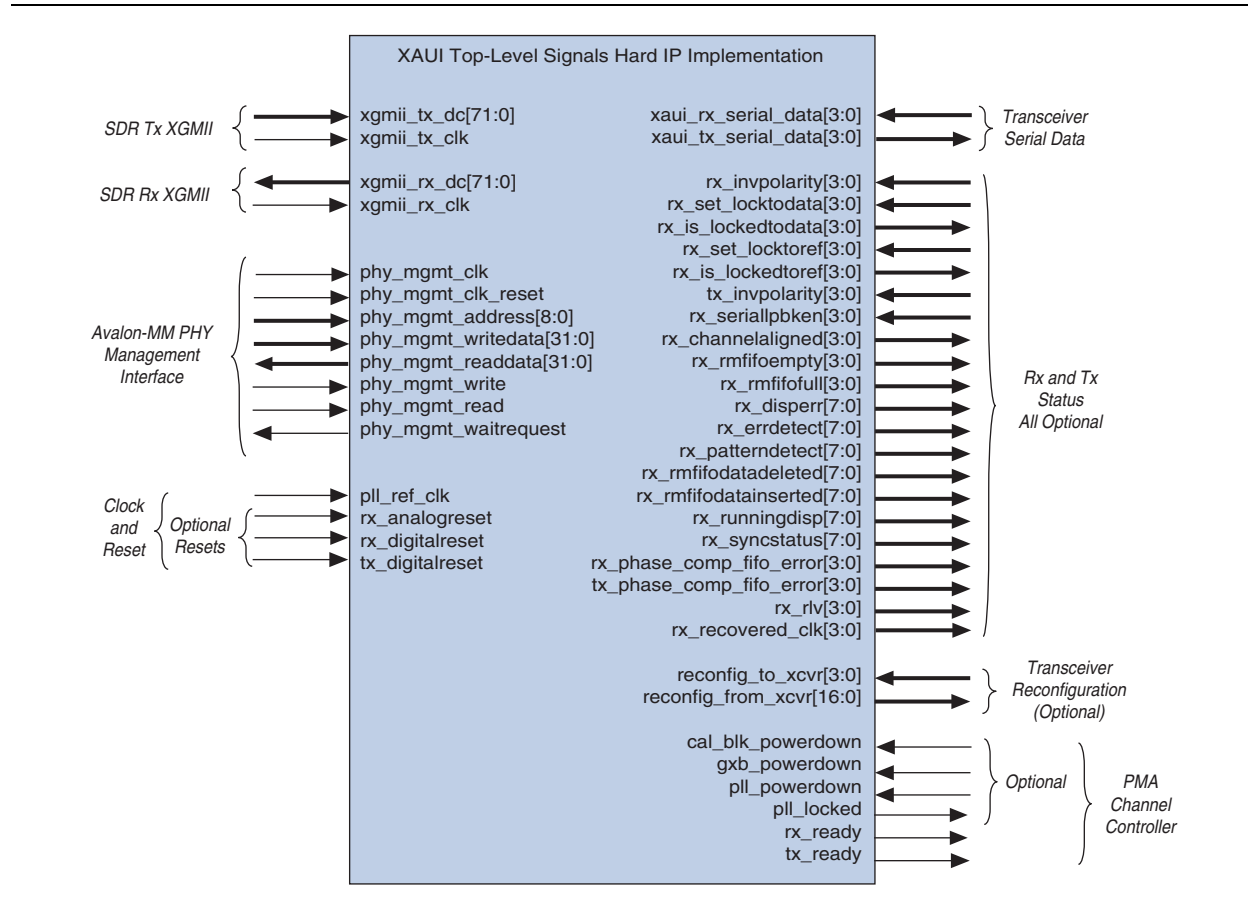
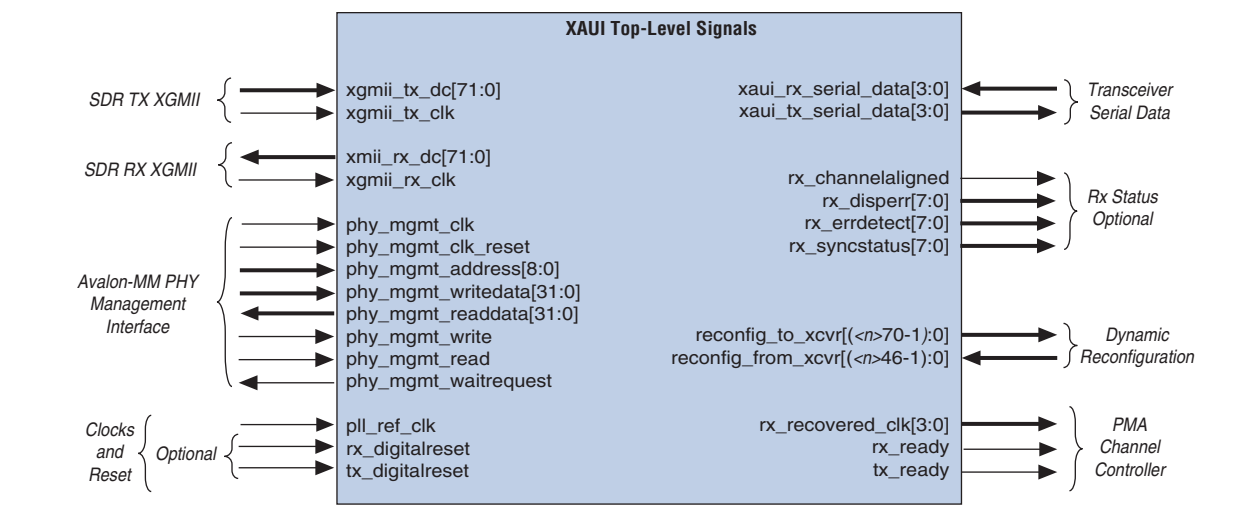


Figure 4-4 illustrates the top-level signals of the XAUI PHY IP core for the soft IP implementation for both the single and DDR rates.

Figure 4-4. XAUI Top-Level Signals—Soft PCS and PMA



The following sections describe the signals in each interface.

SDR XGMII TX Interface

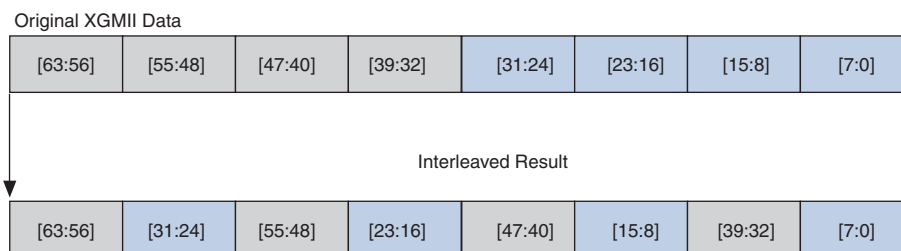
The XAUI PCS interface to the FPGA fabric uses a SDR XGMII interface. This interface implements a simple version of Avalon-ST protocol. The interface does not include ready or valid signals; consequently, the sources always drive data and the sinks must always be ready to receive data.



For more information about the Avalon-ST protocol, including timing diagrams, refer to the [Avalon Interface Specifications](#).

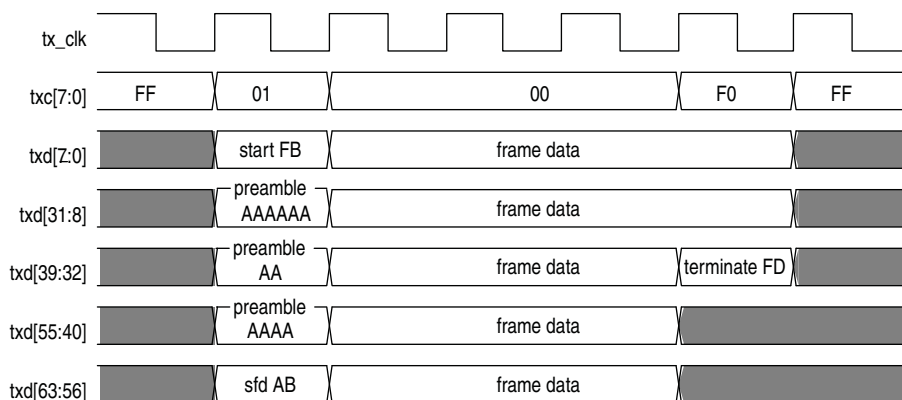
Depending on the parameters you choose, the application interface runs at either 156.25 Mbps or 312.5 Mbps. At either frequency, data is only driven on the rising edge of clock. To meet the bandwidth requirements, the datapath is eight bytes wide with eight control bits, instead of the standard four bytes of data and four bits of control. The XAUI IP core treats the datapath as two, 32-bit data buses and includes logic to interleave them, starting with the low-order bytes. Figure 4-5 illustrates the mapping.

Figure 4-5. Interleaved SDR XGMII Data



For the DDR XAUI variant, the start of control character (0xFB) is aligned to either byte 0 or byte 5. [Figure 4-6](#) illustrates byte 0 alignment.

Figure 4-6. Byte 0 Start of Frame Transmission Example



[Figure 4-7](#) illustrates byte 4 alignment.

Figure 4-7. Byte 4 Start of Frame Transmission Example

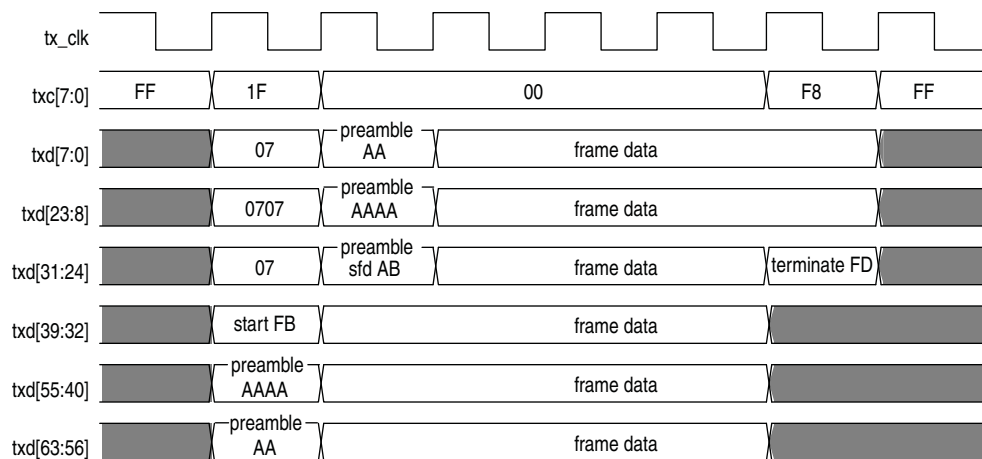


Table 4–9 describes the signals in the SDR TX XGMII interface.

Table 4–9. SDR TX XGMII Interface

Signal Name	Direction	Description
xgmii_tx_dc[71:0]	Source	Contains 4 lanes of data and control for XGMII. Each lane consists of 16 bits of data and 2 bits of control. ■ Lane 0–[7:0]/[8], [43:36]/[44] ■ Lane 1–[16:9]/[17], [52:45]/[53] ■ Lane 2–[25:18]/[26], [61:54]/[62] ■ Lane 3–[34:27]/[35],[70:63]/[71]
xgmii_tx_clk	Input	The XGMII SDR TX clock which runs at 156.25 MHz or 312.5 for the DDR variant.

SDR XGMII RX Interface

Table 4–10 describes the signals in the SDR RX XGMII interface.

Table 4–10. SDR RX XGMII Interface

Signal Name	Direction	Description
xgmii_rx_dc[71:0]	Sink	Contains 4 lanes of data and control for XGMII. Each lane consists of 16 bits of data and 2 bits of control. ■ Lane 0–[7:0]/[8], [43:36]/[44] ■ Lane 1–[16:9]/[17], [52:45]/[53] ■ Lane 2–[25:18]/[26], [61:54]/[62] ■ Lane 3–[34:27]/[35],[70:63]/[71]
xgmii_rx_clk	Output	The XGMII SDR RX MAC interface clock which runs at 156.25 MHz.

Transceiver Serial Data Interface

Table 4–11 describes the signals in the XAUI transceiver serial data interface. There are four lanes of serial data for both the TX and RX interfaces. This interface runs at 3.125 GHz or 6.25 GHz depending on the variant you choose. There is no separate clock signal because it is encoded in the data.

Table 4–11. Serial Data Interface

Signal Name	Direction	Description
xaui_rx_serial_data[3:0]	Input	Serial input data.
xaui_tx_serial_data[3:0]	Output	Serial output data.

Clocks, Reset, and Powerdown

Figure 4-8 illustrates the clock inputs and outputs for the XAUI IP cores with hard PCS and PMA blocks.

Figure 4-8. Clock Inputs and Outputs, Hard PCS

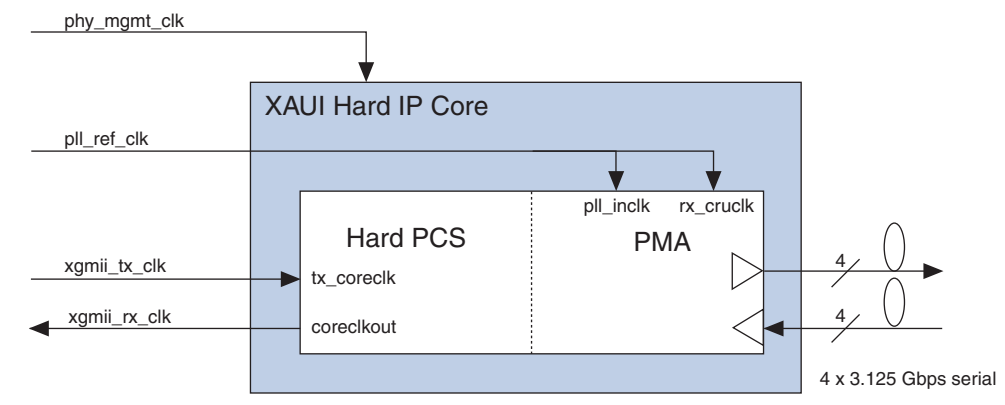


Figure 4-9 illustrates the clock inputs and outputs for the XAUI IP cores with soft PCS and PMA blocks.

Figure 4-9. Clock Inputs and Outputs, Soft PCS

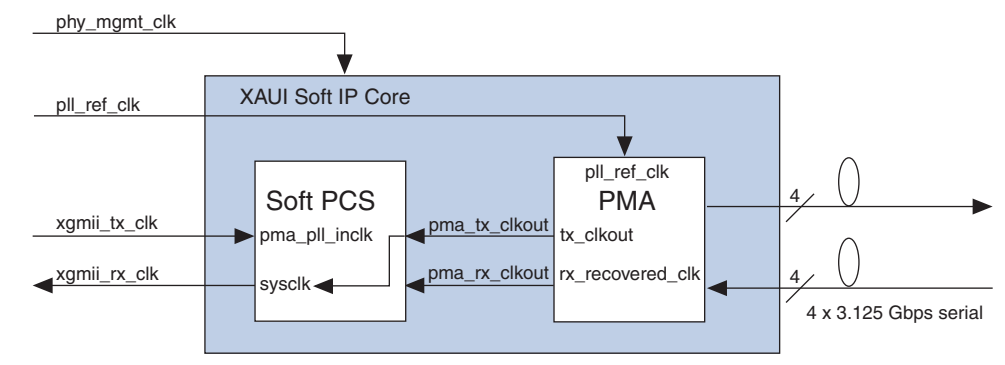


Table 4-12 describes the optional reset signals. Refer to “Reset Controller” on page 1-3 for additional information about reset.

Table 4-12. Clock and Reset Signals

Signal Name	Direction	Description
<code>pll_ref_clk</code>	Input	This is a 156.25 MHz reference clock that is used by the TX PLL and CDR logic.
<code>rx_analogreset</code>	Input	This signal resets the analog CDR and deserializer logic in the RX channel. It is only available in the hard IP implementation.
<code>rx_digitalreset</code>	Input	PCS RX digital reset signal.
<code>tx_digitalreset</code>	Input	PCS TX digital reset signal.

PMA Channel Controller

Table 4-13 describes the signals in this interface.

Table 4-13. PMA Channel Controller Signals

Signal Name	Direction	Description
cal_blk_powerdown	Input	Powers down the calibration block. A high-to-low transition on this signal restarts calibration. Only available in Arria II GX and Stratix IV GX, and Stratix IV GT devices.
gxb_powerdown	Input	When asserted, powers down the entire transceiver block. Only available in Arria II GX and Stratix IV GX, and Stratix IV GT devices.
pll_powerdown	Input	Powers down the CMU PLL. Only available in Arria II GX and Stratix IV GX, and Stratix IV GT devices.
pll_locked	Output	Indicates CMU PLL is locked. Only available in Arria II GX and Stratix IV GX, and Stratix IV GT devices.
rx_recovered_clk[3:0]	Output	This is the RX clock which is recovered from the received data stream.
rx_ready	Output	Indicates PMA RX has exited the reset state and the transceiver can receive data.
tx_ready	Output	Indicates PMA TX has exited the reset state and the transceiver can transmit data.

PMA Control and Status Interface Signals—Soft IP Implementation (Optional)

Table 4-14 lists the optional PMA control and status signals available in the soft IP implementation. You can also access the state of these signals using the Avalon-MM PHY Management interface to read the control and status registers which are detailed in Table 4-17 on page 4-19. However, in some cases, you may need to know the instantaneous value of a signal to ensure correct functioning of the XAUI PHY. In such cases, you can include the required signal in the top-level module of your XAUI PHY IP core.

Table 4-14. Optional Control and Status Signals

Signal Name	Direction	Description
rx_channelaligned	Output	When asserted, indicates that all 4 RX channels are aligned.
rx_disperr[7:0]	Output	Received 10-bit code or data group has a disparity error. It is paired with rx_errdetect which is also asserted when a disparity error occurs. The rx_disperr signal is 2 bits wide per channel for a total of 8 bits per XAUI link.
rx_errdetect[7:0]	Output	When asserted, indicates an 8B/10B code group violation. It is asserted if the received 10-bit code group has a code violation or disparity error. It is used along with the rx_disperr signal to differentiate between a code violation error, a disparity error, or both. The rx_errdetect signal is 2 bits wide per channel for a total of 8 bits per XAUI link.
rx_syncstatus[7:0]	Output	Synchronization indication. RX synchronization is indicated on the rx_syncstatus port of each channel. The rx_syncstatus signal is 2 bits per channel for a total of 8 bits per hard XAUI link. The rx_syncstatus signal is 1 bit per channel for a total of 4 bits per soft XAUI link.

Table 4-14. Optional Control and Status Signals

Signal Name	Direction	Description
rx_is_lockedtodata[3:0]	Output	When asserted indicates that the RX CDR PLL is locked to the incoming data.
rx_is_lockedtoref[3:0]	Output	When asserted indicates that the RX CDR PLL is locked to the reference clock.
tx_clk312_5	Output	This is the clock used for the SDR XGMII interface.

PMA Control and Status Interface Signals—Hard IP Implementation (Optional)

Table 4-15 lists the PMA control and status signals. You can access the state of these signals using the Avalon-MM PHY Management interface to read the control and status registers which are detailed in Table 4-17 on page 4-19. However, in some cases, you may need to know the instantaneous value of a signal to ensure correct functioning of the XAUI PHY. In such cases, you can include the required signal in the top-level module of your XAUI PHY IP core.

Table 4-15. Optional Control and Status Signals—Hard IP Implementation, Stratix IV GX Devices (Part 1 of 2)

Signal Name	Direction	Description
rx_invpolarity[3:0]	input	Dynamically reverse the polarity of every bit of the RX data at the input of the word aligner.
rx_set_locktodata[3:0]	Input	Force the CDR circuitry to lock to the received data.
rx_is_lockedtodata[3:0]	Output	When asserted, indicates the RX channel is locked to input data.
rx_set_locktoref[3:0]	Input	Force the receiver CDR to lock to the phase and frequency of the input reference clock.
rx_is_lockedtoref[3:0]	Output	When asserted, indicates the RX channel is locked to input reference clock.
tx_invpolarity[3:0]	input	Dynamically reverse the polarity the data word input to the serializer in the TX datapath.
rx_serialpbken	input	Serial loopback enable. 1: Enables serial loopback 0: Disables serial loopback This signal is asynchronous to the receiver. The status of the serial loopback option is recorded by the PMA channel controller, word address 0x061.
rx_channelaligned	Output	When asserted indicates that the RX channel is aligned.
pll_locked	Output	In LTR mode, indicates that the receiver CDR has locked to the phase and frequency of the input reference clock.
rx_rmfifoempty[3:0]	Output	Status flag that indicates the rate match FIFO block is empty (5 words). This signal remains high as long as the FIFO is empty and is asynchronous to the RX datapath.
rx_rmfull[3:0]	Output	Status flag that indicates the rate match FIFO block is full (20 words). This signal remains high as long as the FIFO is full and is asynchronous to the RX data.
rx_disperr[7:0]	Output	Received 10-bit code or data group has a disparity error. It is paired with rx_errdetect which is also asserted when a disparity error occurs. The rx_disperr signal is 2 bits wide per channel for a total of 8 bits per XAUI link.

Table 4-15. Optional Control and Status Signals—Hard IP Implementation, Stratix IV GX Devices (Part 2 of 2)

Signal Name	Direction	Description
<code>rx_errdetect[7:0]</code>	Output	Transceiver 8B/10B code group violation or disparity error indicator. If either signal is asserted, a code group violation or disparity error was detected on the associated received code group. Use the <code>rx_disperr</code> signal to determine whether this signal indicates a code group violation or a disparity error. The <code>rx_errdetect</code> signal is 2 bits wide per channel for a total of 8 bits per XAUI link.
<code>rx_patterndetect[7:0]</code>	Output	Indicates that the word alignment pattern programmed has been detected in the current word boundary. The <code>rx_patterndetect</code> signal is 2 bits wide per channel for a total of 8 bits per XAUI link.
<code>rx_rmifodatadeleted[7:0]</code>	Output	Status flag that is asserted when the rate match block deletes a <code> R </code> column. The flag is asserted for one clock cycle per deleted <code> R </code> column.
<code>rx_rmifodatainserted[7:0]</code>	Output	Status flag that is asserted when the rate match block inserts a <code> R </code> column. The flag is asserted for one clock cycle per inserted <code> R </code> column.
<code>rx_runningdisp[7:0]</code>	Output	Asserted when the current running disparity of the 8B/10B decoded byte is negative. Low when the current running disparity of the 8B/10B decoded byte is positive.
<code>rx_syncstatus[7:0]</code>	Output	Synchronization indication. RX synchronization is indicated on the <code>rx_syncstatus</code> port of each channel. The <code>rx_syncstatus</code> signal is 2 bits wide per channel for a total of 8 bits per XAUI link.
<code>rx_phase_comp_fifo_error[3:0]</code>	Output	Indicates a RX phase comp FIFO overflow or underrun condition.
<code>tx_phase_comp_fifo_error[3:0]</code>	Output	Indicates a TX phase compensation FIFO overflow or underrun condition.
<code>rx_rlv[3:0]</code>	Output	Asserted if the number of continuous 1s and 0s exceeds the number that was set in the run-length option. The <code>rx_rlv</code> signal is asynchronous to the RX datapath and is asserted for a minimum of 2 recovered clock cycles.
<code>rx_recovered_clk</code>	Output	This is the RX clock which is recovered from the received data stream.

Registers

The Avalon-MM PHY management interface provides access to the XAUI PHY IP Core PCS, PMA, and transceiver reconfiguration registers. [Table 4-16](#) describes the signals that comprise the Avalon-MM PHY Management interface.

Table 4-16. Avalon-MM PHY Management Interface

Signal Name	Direction	Description
phy_mgmt_clk	Input	Avalon-MM clock input. There is no frequency restriction for Stratix V devices; however, if you plan to use the same clock for the PHY management interface and transceiver reconfiguration, you must restrict the frequency range of phy_mgmt_clk to 100–125 MHz to meet the specification for the transceiver reconfiguration clock. For Arria II GX, Cyclone IV GX, HardCopy IV, and Stratix IV GX the frequency range is 37.5–50 MHz.
phy_mgmt_clk_reset	Input	Global reset signal that resets the entire XAUI PHY. This signal is active high and level sensitive.
phy_mgmt_addr[8:0]	Input	9-bit Avalon-MM address.
phy_mgmt_writedata[31:0]	Input	32-bit input data.
phy_mgmt_readdata[31:0]	Output	32-bit output data.
phy_mgmt_write	Input	Write signal. Asserted high.
phy_mgmt_read	Input	Read signal. Asserted high.
phy_mgmt_waitrequest	Output	When asserted, indicates that the Avalon-MM slave interface is unable to respond to a read or write request. When asserted, control signals to the Avalon-MM slave interface must remain constant.



For more information about the Avalon-MM interface, including timing diagrams, refer to the [Avalon Interface Specifications](#).

Register Descriptions

[Table 4-17](#) specifies the registers that you can access using the Avalon-MM PHY management interface using word addresses and a 32-bit embedded processor. A single address space provides access to all registers.



Writing to reserved or undefined register addresses may have undefined side effects.

Table 4-17. XAUI PHY IP Core Registers (Part 1 of 5)

Word Addr	Bits	R/W	Register Name	Description
PMA Common Control and Status Registers				
0x021	[31:0]	RW	cal_blk_powerdown	Writing a 1 to channel <i><n></i> powers down the calibration block for channel <i><n></i> .
0x022	[31:0]	R	pma_tx_pll_is_locked	Bit[P] indicates that the TX CMU PLL (P) is locked to the input reference clock. There is typically one pma_tx_pll_is_locked bit per system. This register is not available for Stratix V devices.

Table 4–17. XAUI PHY IP Core Registers (Part 2 of 5)

Word Addr	Bits	R/W	Register Name	Description
Reset Control Registers—Automatic Reset Controller				
0x041	[31:0]	RW	reset_ch_bitmask	Bit mask for reset registers at addresses 0x042 and 0x044. The default value is all 1s. Channel $\langle n \rangle$ can be reset when bit $\langle n \rangle$ = 1.
0x042	[1:0]	W	reset_control (write)	Writing a 1 to bit 0 initiates a TX digital reset using the reset controller module. The reset affects channels enabled in the reset_ch_bitmask. Writing a 1 to bit 1 initiates a RX digital reset of channels enabled in the reset_ch_bitmask.
		R	reset_status(read)	Reading bit 0 returns the status of the reset controller TX ready bit. Reading bit 1 returns the status of the reset controller RX ready bit.
Reset Controls —Manual Mode				
0x044	[31:4,0]	RW	Reserved	It is safe to write 0s to reserved bits.
	[1]	RW	reset_tx_digital	Writing a 1 causes the internal TX digital reset signal to be asserted, resetting all channels enabled in reset_ch_bitmask. You must write a 0 to clear the reset condition.
	[2]	RW	reset_rx_analog	Writing a 1 causes the internal RX analog reset signal to be asserted, resetting the RX analog logic of all channels enabled in reset_ch_bitmask. You must write a 0 to clear the reset condition.
	[3]	RW	reset_rx_digital	Writing a 1 causes the RX digital reset signal to be asserted, resetting the RX digital channels enabled in reset_ch_bitmask. You must write a 0 to clear the reset condition.
PMA Control and Status Registers				
0x061	[31:0]	RW	phy_serial_loopback	Writing a 1 to channel $\langle n \rangle$ puts channel $\langle n \rangle$ in serial loopback mode. For information about pre- or post-CDRserial loopback modes, refer to “ Loopback Modes ” on page 10–39 .
0x064	[31:0]	RW	pma_rx_set_locktodata	When set, programs the RX CDR PLL to lock to the incoming data. Bit $\langle n \rangle$ corresponds to channel $\langle n \rangle$.
0x065	[31:0]	RW	pma_rx_set_locktoref	When set, programs the RX CDR PLL to lock to the reference clock. Bit $\langle n \rangle$ corresponds to channel $\langle n \rangle$.
0x066	[31:0]	R	pma_rx_is_lockedtodata	When asserted, indicates that the RX CDR PLL is locked to the RX data, and that the RX CDR has changed from LTR to LTD mode. Bit $\langle n \rangle$ corresponds to channel $\langle n \rangle$.
0x067	[31:0]	R	pma_rx_is_lockedtoref	When asserted, indicates that the RX CDR PLL is locked to the reference clock. Bit $\langle n \rangle$ corresponds to channel $\langle n \rangle$.

Table 4–17. XAUI PHY IP Core Registers (Part 3 of 5)

Word Addr	Bits	R/W	Register Name	Description
XAUI PCS				
0x082	[31:4]	—	Reserved	—
	[3:0]	RW	invpolarity[3:0]	Inverts the polarity of corresponding bit on the RX interface. Bit 0 maps to lane 0 and so on. To block: Word aligner.
0x083	[31:4]	—	Reserved	—
	[3:0]	RW	invpolarity[3:0]	Inverts the polarity of corresponding bit on the TX interface. Bit 0 maps to lane 0 and so on. To block: Serializer.
0x084	[31:16]	—	Reserved	—
	[15:8]	R	patterndetect[7:0]	When asserted, indicates that the programmed word alignment pattern has been detected in the current word boundary. The RX pattern detect signal is 2 bits wide per channel or 8 bits per XAUI link. Reading the value of the <code>patterndetect</code> registers clears the bits. This register is only available in the hard XAUI implementation. From block: Word aligner.
	[7:0]		syncstatus[7:0]	Records the synchronization status of the corresponding bit. The RX sync status register has 2 bits per channel for a total of 8 bits per hard XAUI link. The RX sync status register has 1 bit per channel for a total of 4 bits per soft XAUI link; soft XAUI uses bits 0–3. Reading the value of the <code>syncstatus</code> register clears the bits. From block: Word aligner.
0x085	[31:16]	—	Reserved	—
	[15:8]	R	errdetect[7:0]	When set, indicates that a received 10-bit code group has an 8B/10B code violation or disparity error. It is used along with <code>disperr</code> to differentiate between a code violation error, a disparity error, or both. There are 2 bits per RX channel for a total of 8 bits per XAUI link. Reading the value of the <code>errdetect</code> register clears the bits. From block: 8B/10B decoder.
	[7:0]		disperr[7:0]	Indicates that the received 10-bit code or data group has a disparity error. When set, the corresponding <code>errdetect</code> bits are also set. There are 2 bits per RX channel for a total of 8 bits per XAUI link. Reading the value of the <code>errdetect</code> register clears the bits From block: 8B/10B decoder.

Table 4–17. XAUI PHY IP Core Registers (Part 4 of 5)

Word Addr	Bits	R/W	Register Name	Description
0x086	[31:8]	—	Reserved	—
	[7:4]	R, sticky	phase_comp_fifo_error[3:0]	Indicates a RX phase compensation FIFO overflow or underrun condition on the corresponding lane. Reading the value of the phase_comp_fifo_error register clears the bits. This register is only available in the hard XAUI implementation From block: RX phase compensation FIFO.
	[3:0]		rlv[3:0]	Indicates a run length violation. Asserted if the number of consecutive 1s or 0s exceeds the number that was set in the Runlength check option. Bits 0-3 correspond to lanes 0-3, respectively. Reading the value of the RLV register clears the bits. This register is only available in the hard XAUI implementation. From block: Word aligner.
0x087	[31:16]	—	Reserved	—
	[15:8]	R, sticky	rmfifodatainserted[7:0]	When asserted, indicates that the RX rate match block inserted a R column. Goes high for one clock cycle per inserted R column. Reading the value of the rmfifodatainserted register clears the bits. This register is only available in the hard XAUI implementation. From block: Rate match FIFO.
	[7:0]		rmfifodatadeleted[7:0]	When asserted, indicates that the rate match block has deleted an R column. The flag goes high for one clock cycle per deleted R column. There are 2 bits for each lane. Reading the value of the rmfifodatadeleted register clears the bits. This register is only available in the hard XAUI implementation. From block: Rate match FIFO.
0x088	[31:8]	—	Reserved	—
	[7:4]	R, sticky	rmfifofull[3:0]	When asserted, indicates that rate match FIFO is full (20 words). Bits 0-3 correspond to lanes 0-3, respectively. Reading the value of the rmfifofull register clears the bits. This register is only available in the hard XAUI implementation From block: Rate match FIFO.
	[3:0]		rmfifoepty[3:0]	When asserted, indicates that the rate match FIFO is empty (5 words). Bits 0-3 correspond to lanes 0-3, respectively. Reading the value of the rmfifoepty register clears the bits. This register is only available in the hard XAUI implementation From block: Rate match FIFO.

Table 4–17. XAUI PHY IP Core Registers (Part 5 of 5)

Word Addr	Bits	R/W	Register Name	Description
0x089	[31:3]	—	Reserved	—
	[2:0]	R, sticky	phase_comp_fifo_error[2:0]	Indicates a TX phase compensation FIFO overflow or underrun condition on the corresponding lane. Reading the value of the phase_comp_fifo_error register clears the bits. This register is only available in the hard XAUI implementation From block: TX phase compensation FIFO.
0x08a	[0]	RW	simulation_flag	Setting this bit to 1 shortens the duration of reset and loss timer when simulating. Altera recommends that you keep this bit set during simulation.

Dynamic Reconfiguration

As silicon progresses towards smaller process nodes, circuit performance is affected more by variations due to process, voltage, and temperature (PVT). These process variations result in analog voltages that can be offset from required ranges. The calibration performed by the dynamic reconfiguration interface compensates for variations due to PVT.

This section includes information about the dynamic reconfiguration interface. The Arria II GX, Cyclone IV GX, HardCopy IV GX, and Stratix IV GX use the ALTGX_RECONFIG Megafunction for transceiver reconfiguration. The Stratix V device uses the Transceiver Reconfiguration Controller IP core for dynamic reconfiguration. For more information about this IP core, refer to [Chapter 10, Transceiver Reconfiguration Controller](#).



For more information about the ALTGX_RECONFIG Megafunction, refer to *ALTGX_RECONFIG Megafunction User Guide for Stratix IV Devices* in volume 2 of the *Stratix IV Device Handbook*.

Reconfiguration for Arria II GX, Cyclone IV GX, HardCopy IV GX, and Stratix IV GX Devices

[Table 4–18](#) describes the signals in the reconfiguration interface. If your XAUI PHY IP core includes a single transceiver quad, these signals are internal to the core. If your design uses more than one quad, the reconfiguration signals are external.

Table 4–18. Dynamic Reconfiguration Interface

Signal Name	Direction	Description
reconfig_to_xcvr[3:0]	Input	Reconfiguration signals from the Transceiver Reconfiguration IP core to the XAUI transceiver.
reconfig_from_xcvr[<n>:0]	Output	Reconfiguration signals from the XAUI transceiver to the Transceiver Reconfiguration IP core. The size of this bus is depends on the device. For the soft PCS in Stratix IV GX and GT devices, <n> = 68 bits. For hard XAUI variants, <n> = 16. For Stratix V devices, the number of bits depends on the number of channels specified. Refer to Chapter 10, Transceiver Reconfiguration Controller for more information.

Dynamic Reconfiguration for Stratix V Devices

For Stratix V devices, each channel and each TX PLL have separate dynamic reconfiguration interfaces. The parameter editor provides informational messages on the connectivity of these interfaces. [Example 4-1](#) shows the messages for a single transceiver quad.

Although you must initially create a separate reconfiguration interface for each channel and TX PLL in your design, when the Quartus II software compile your design, it reduces the number of reconfiguration interfaces by merging reconfiguration interfaces. The synthesized design typically includes a reconfiguration interface for three channels. Allowing the Quartus II software to merge reconfiguration interfaces gives the Fitter more flexibility in placing transceiver channels.

Example 4-1. Informational Messages for the Transceiver Reconfiguration Interface

```
PHY IP will require 8 reconfiguration interfaces for connection to the external
reconfiguration controller.
Reconfiguration interface offsets 0-3 are connected to the transceiver channels.
Reconfiguration interface offsets 4-7 are connected to the transmit PLLs.
```

Simulation Files and Example Testbench

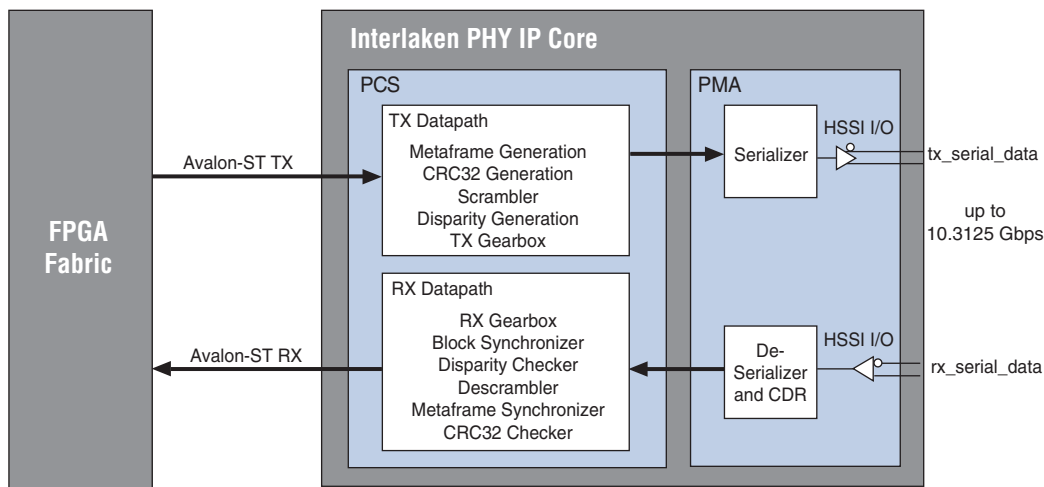
Refer to [“Running a Simulation Testbench” on page 1-4](#) for a description of the directories and files that the Quartus II software creates automatically when you generate your XAUI PHY IP core.



Refer to the [Altera wiki](#) for an example testbench that you can use as a starting point in creating your own verification environment.

Interlaken is a high speed serial communication protocol for chip-to-chip packet transfers. The Altera Interlaken PHY IP core implements *Interlaken Protocol Specification, Rev 1.2*. It supports multiple instances, each with 1 to 24 lanes running at up to 10.3125 Gbps on Stratix V devices. The key advantages of Interlaken are scalability and its low I/O count compared to earlier protocols such as SPI 4.2. Other key features include flow control, low overhead framing, and extensive integrity checking. The Interlaken physical coding sublayer (PCS) transmits and receives Avalon-ST data on its FPGA fabric interface. It transmits and receives high speed differential serial data using the PCML I/O standard. Figure 5–1 illustrates the top-level modules of the Interlaken PHY.

Figure 5–1. Interlaken PHY IP Core



For more information about the Avalon-ST protocol, including timing diagrams, refer to the *Avalon Interface Specifications*.

Interlaken operates on 64-bit data words and 3 control bits, which are striped round robin across the lanes to reduce latency. Striping renders the interface independent of exact lane count. The protocol accepts packets on 256 logical channels and is expandable to accommodate up to 65,536 logical channels. Packets are split into small bursts which can optionally be interleaved. The burst semantics include integrity checking and per channel flow control.

The Interlaken PCS supports the following framing functions on a per lane basis:

- Gearbox
- Block synchronization
- Metaframe generation and synchronization
- 64b/67b encoding and decoding
- Scrambling and descrambling

- Lane-based CRC32
 - Disparity DC balancing
- For more detailed information about the Interlaken transceiver channel datapath, clocking, and channel placement, refer to the “Interlaken” section in the *Transceiver Protocol Configurations in Stratix V Devices* chapter of the *Stratix V Device Handbook*.
- Refer to *PHY IP Design Flow with Interlaken for Stratix V Devices* for a reference design that implements the Interlaken protocol in a Stratix V device.

Device Family Support

IP cores provide either final or preliminary support for target Altera device families. These terms have the following definitions:

- *Final support*—Verified with final timing models for this device.
- *Preliminary support*—Verified with preliminary timing models for this device.

Table 5–1 shows the level of support offered by the Interlaken PHY IP core for Altera device families

Table 5–1. Device Family Support

Device Family	Support
Stratix V devices—Hard PCS + PMA	Preliminary
Other device families	Not supported

Parameter Settings

To configure the Interlaken PHY IP core in the parameter editor, click **Installed Plug-Ins > Interfaces > Interlaken > Interlaken PHY v11.1**. The Interlaken PHY IP core is only available when you select the **Stratix V** device family.

General Options

Table 5–2 describes the parameters that you can set on the **General** tab.

Table 5–2. General Option (Part 1 of 2)

Parameter	Value	Description
Device family	Stratix V	Specifies the device family.
Datapath mode	Duplex, RX, TX	Specifies the mode of operation as Duplex , RX , or TX mode.
Lane rate	3125 Mbps 5000 Mbps 6250 Mbps 6375 Mbps 10312.5 Mbps	Specifies the link bandwidth. The Input clock frequency and Base data rate parameters update automatically based on the Lane rate you specify; however, you can change these default values.
Number of lanes	1–24	Specifies the number of lanes in a link over which data is striped.
Metaframe length in words	5–8191	Specifies the number of words in a metaframe. The default value is 2048.

Table 5–2. General Option (Part 2 of 2)

Parameter	Value	Description
Input clock frequency	Lane rate/<n> Lane rate/80 Lane rate/64 Lane rate/50 Lane rate/40 Lane rate/32 Lane rate/25 Lane rate/20 Lane rate/16 Lane rate/12.5 Lane rate/10 Lane rate/8	Specifies the frequency of the input reference clock. The default value for the Input clock frequency is the Lane rate /20; however, you can change this value. Many reference clock frequencies are available.
PLL type	CMU ATX	Specifies the PLL type. The CMU PLL has a larger frequency range than the ATX PLL. The ATX PLL is designed to improve jitter performance and achieves lower channel-to-channel skew; however, it supports a narrower range of data rates and reference clock frequencies. Another advantage of the ATX PLL is that it does not use a transceiver channel, while the CMU PLL does. Because the CMU PLL is more versatile, it is specified as the default setting.
Base data rate	1 × Lane rate 2 × Lane rate 4 × Lane rate	This option allows you to specify a Base data rate to minimize the number of PLLs required to generate the clocks necessary for data transmission at different frequencies. Depending on the Lane rate you specify, the default Base data rate can be either 1, 2, or 4 times the Lane rate ; however, you can change this value. The default value specified is for backwards compatibility with earlier Quartus II software releases.

Advanced Options

Table 5–2 describes the parameters that you can set on the **Optional Ports** tab.

Table 5–3. Optional Ports

Parameter	Value	Description
Enable RX status signals, (word lock, sync lock, crc32 error) as part of rx_parallel_data	On/Off	When you turn this option on, rx_parallel_data[71:69] are included in the top-level module. These optional signals report the status of word and synchronization locks and CRC32 errors. Refer to Table 5–7 on page 5–10 for more information. facilitate
Create tx_coreclkkin port	On/Off	The tx_coreclkkin drives the write side of TX FIFO. This clock is required for lane synchronization. tx_coreclkkin must be used when the number of lanes is greater than 1.
Create rx_coreclkkin port	On/Off	When selected rx_coreclkkin is available as input port which drives the read side of RX FIFO, When deselected rx_user_clkout, which is the master rx_clockout for all bonded receiver lanes, is routed internally to drive the RX read side of FIFO. rx_user_clkout is also available as an output port for the Interlaken MAC.

Analog Settings

You specify the analog parameters for Stratix V devices using the Quartus II Assignment Editor, the Pin Planner, or through the Quartus II Settings File (.qsf). The default values for analog options fall into three categories:

- *Global*—These parameters have default values that are independent of other parameter settings.
- *Computed*—These parameters have an initial default value that is recomputed based on other parameter settings.
- *Proxy*—These parameters have default values that are place holders. The Quartus II software selects these initial default values based on your design; however, Altera recommends that you replace these defaults with values that match your electrical board specification.

Table 5-4 lists the analog parameters for Stratix V devices whose original values are place holders for the values that match your electrical board specification. In Table 5-4, the default value of an analog parameter is shown in **bold** type.

Table 5-4. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 2)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_IO_PIN_TERMINATION	Transceiver I/O Pin Termination	Specifies the intended on-chip termination value for the specified transceiver pin. Use External Resistor if you intend to use off-chip termination.	85_OHMS 100_OHMS 120_OHMS 150_OHMS EXTERNAL_RESISTOR	Pin
XCVR_REFCLK_PIN_TERMINATION	Transceiver Dedicated Refclk Pin Termination	Specifies the intended termination value for the specified refclk pin.	DC_COUPLING_INTERNAL_100_OHM DC_COUPLING_EXTERNAL_RESISTOR AC_COUPLING	Pin
XCVR_RX_BYPASS_EQ_STAGES_234	Receiver Equalizer Stage 2, 3, 4 Bypass	Bypass continuous time equalizer stages 2, 3, and 4 to save power. This setting eliminates significant AC gain on the equalizer and is appropriate for chip-to-chip short range communication on a PCB.	ALL_STAGES_ENABLED BYPASS_STAGES	Pin
XCVR_TX_SLEW_RATE_CTRL	Transmitter Slew Rate Control	Specifies the slew rate of the output signal. The valid values span from the slowest rate to fastest rate with 1 representing the slowest rate.	1-5	Pin

Table 5–4. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 2)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_VCCA_VOLTAGE	VCCA_GXB Voltage	Configure the VCCA_GXB voltage for a GXB I/O pin by specifying the intended VCCA_GXB voltage for a GXB I/O pin. If you do not make this assignment the compiler automatically sets the correct VCCA_GXB voltage depending on the configured data rate, as follows: ■ Data rate <= 6.5 Gbps: 2_5V ■ Data rate > 6.5 Gbps: 3_0V or 3_3V for Stratix V ES silicon	2_5V 3_0V	Pin
XCVR_VCCR_VCCT_VOLTAGE	VCCR_GXB VCCT_GXB Voltage	Configure the VCCR_GXB and VCCT_GXB voltage for an GXB I/O pin by specifying the intended supply voltages for a GXB I/O pin. If your design uses decision feedback equalization (DFE) or adaptive equalization (AEQ), you must set this parameter to 1.0V. Otherwise, if you do not make this assignment the compiler automatically sets the correct VCCR_GXB and VCCT_GXB voltage depending on the configured data rate as follows: ■ Data rate <= 6.5 Gbps: 0_85V ■ Data rate > 6.5 Gbps: 1_0V	0_85V 1_0V	Pin

Table 5–5 lists the analog parameters with *global* or *computed* default values. You may want to optimize some of these settings. In Table 5–5, the default value is shown in **bold** type. For computed analog parameters, the default value listed is for the initial setting, not the recomputed setting.

Table 5–5. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 3)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
Analog Parameters with Global Default Value				
CDR_BANDWIDTH_PRESET	CDR Bandwidth Preset	Specifies the CDR bandwidth preset setting.	Auto Low Medium High	PLL instance
PLL_BANDWIDTH_PRESET	PLL Bandwidth Preset	Specifies the PLL bandwidth preset setting	Auto Low Medium High	PLL instance
XCVR_RX_DC_GAIN	Receiver Buffer DC Gain Control	Controls the amount of a stage receive-buffer DC gain.	0–4	Pin

Table 5–5. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 3)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_RX_LINEAR_EQUALIZER_CONTROL	Receiver Linear Equalizer Control	Static control for the continuous time equalizer in the receiver buffer. The equalizer has 16 distinct settings from 0–15 corresponding to the increasing AC gain.	1–16	Pin
Analog Parameter with Computed Default Value				
XCVR_RX_COMMON_MODE_VOLTAGE	Receiver Buffer Common Mode Voltage	Receiver buffer common-mode voltage.	VTT_OP80V VTT_OP75V VTT_OP70V VTT_OP65V VTT_OP60V VTT_OP55V VTT_OP50V VTT_OP35V VTT_PUP_WEAK VTT_PDN_WEAK TRISTATE1 VTT_PDN_STRONG VTT_PUP_STRONG TRISTATE2 TRISTATE3 TRISTATE4	Pin
XCVR_RX_ENABLE_LINEAR_EQUALIZER_PCIEMODE	Receiver Linear Equalizer Control (PCI Express)	If enabled equalizer gain control is driven by the PCS block for PCI Express. If disabled equalizer gain control is determined by the XCVR_RX_LINEAR_EQUALIZER_SETTING assignment.	TRUE FALSE	Pin
XCVR_RX_EQ_BW_SEL	Receiver Equalizer Gain Bandwidth Select	Sets the gain peaking frequency for the equalizer. For data-rates of less than 6.5Gbps set to HALF. For higher data-rates set to FULL.	FULL HALF	Pin
XCVR_RX_SD_ENABLE	Receiver Signal Detection Unit Enable/Disable	Enables or disables the receiver signal detection unit.	TRUE FALSE	Pin
XCVR_RX_SD_OFF	Receiver Cycle Count Before Signal Detect Block Declares Loss Of Signal	Number of parallel cycles to wait before the signal detect block declares loss of signal.	0 –29	Pin
XCVR_RX_SD_ON	Receiver Cycle Count Before Signal Detect Block Declares Presence Of Signal	Number of parallel cycles to wait before the signal detect block declares presence of signal.	0 –16	Pin
XCVR_RX_SD_THRESHOLD	Receiver Signal Detection Voltage Threshold	Specifies signal detection voltage threshold level.	0 –7	Pin

Table 5–5. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 3)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_TX_COMMON_MODE_VOLTAGE	Transmitter Common Mode Driver Voltage	Transmitter common-mode driver voltage	VOLT_OP80V VOLT_OP75V VOLT_OP70V VOLT_OP65V VOLT_OP60V VOLT_OP55V VOLT_OP50V VOLT_OP35V PULL_UP PULL_DOWN TRISTATED1 GROUNDED PULL_UP_TO_VCCELA TRISTATED2 TRISTATED3 TRISTATED4	Pin
XCVR_TX_PRE_EMP_1ST_POST_TAP	Transmitter Preemphasis First Post-Tap	Specifies the first post-tap setting value.	0 –31	Pin
XCVR_TX_PRE_EMP_2ND_POST_TAP	Transmitter Preemphasis Second Post-Tap	Specifies the second post-tap setting value.	0 –15	Pin
XCVR_TX_PRE_EMP_INV_2ND_TAP	Transmitter Preemphasis Second Tap Invert	Inverts the transmitter pre-emphasis 2nd post tap.	TRUE FALSE	Pin
XCVR_TX_PRE_EMP_INV_PRE_TAP	Transmitter Preemphasis Pre Tap Invert	Inverts the transmitter pre-emphasis pre-tap.	TRUE FALSE	Pin
XCVR_TX_PRE_EMP_PRE_TAP	Transmitter Preemphasis Pre-Tap	Specifies the pre-tap pre-emphasis setting.	0 –15	Pin
XCVR_TX_RX_DET_ENABLE	Transmitter's Receiver Detect Block Enable	Enables or disables the receiver detector circuit at the transmitter.	TRUE FALSE	Pin
XCVR_TX_RX_DET_MODE	Transmitter's Receiver Detect Block Mode	Sets the mode for receiver detect block	0 –15	Pin
XCVR_TX_RX_DET_OUTPUT_SEL	Transmitter's Receiver Detect Block QPI/PCI Express Control	Determines QPI or PCI Express mode for the Receiver Detect block.	RX_DET_QPI_OUT RX_DET_PCIE_OUT	Pin
XCVR_TX_VOD	Transmitter Differential Output Voltage	Differential output voltage setting. The values are monotonically increasing with the driver main tap current strength.	0–63 50	Pin
XCVR_TX_VOD_PRE_EMP_CTRL_SRC	Transmitter V _{OD} /Preemphasis Control Source	When set to DYNAMIC_CTL, the PCS block controls the V _{OD} and preemphasis coefficients for PCI Express. When this assignment is set to RAM_CTL the V _{OD} and preemphasis are controlled by other assignments, such as XCVR_TX_PRE_EMP_1ST_POST_TAP.	DYNAMIC_CTL RAM_CTL	Pin

For more information about the Pin Planner, refer to [About the Pin Planner](#) in Quartus II Help. For more information about the Assignment Editor, refer to [About the Assignment Editor](#) in Quartus II Help.

For more information about Quartus II Settings, refer to [Quartus II Settings File Manual](#).

Interfaces

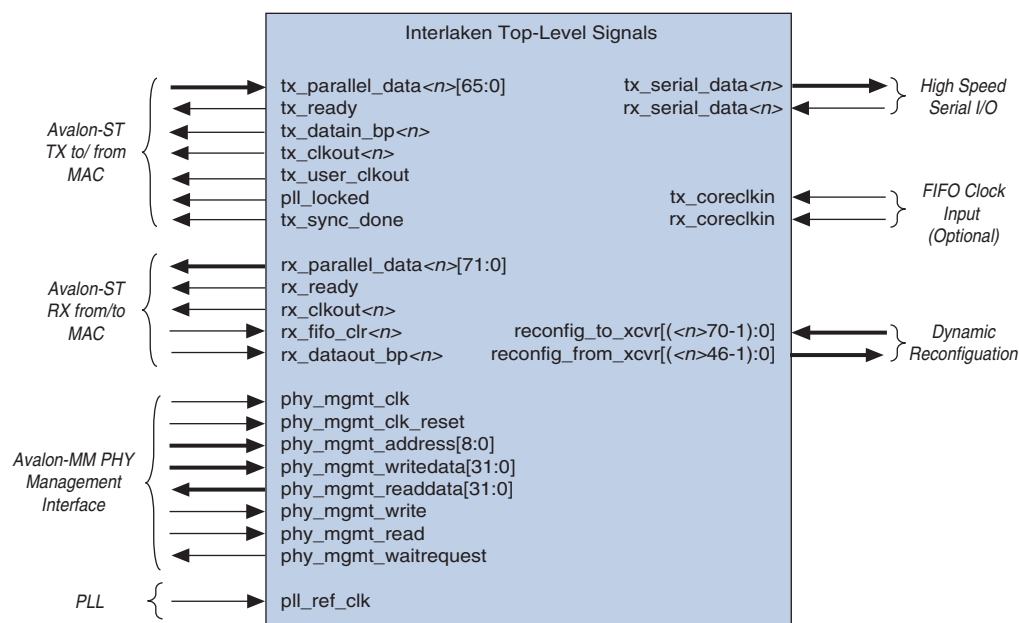
This section describes interfaces of the Interlaken Transceiver PHY. It includes the following sections:

- [Ports](#)
- [Registers](#)
- [Transceiver Reconfiguration](#)

Ports

Figure 5-2 illustrates the top-level signals of the Interlaken PHY IP core.

Figure 5-2. Top-Level Interlaken PHY Signals ⁽¹⁾



Note to Figure 5-2:

(1) <n> = the number of channels in the interface, so that the width of tx_data in 4-channel instantiation is [263:0].



The **block diagram** shown in the GUI labels the external pins with the *interface type* and places the *interface name* inside the box. The interface type and name are used to define interfaces in the `_hw.tcl` writing



For more information about `_hw.tcl`, files refer to the *Component Interface Tcl Reference* chapter in volume 1 of the *Quartus II Handbook*.

The following sections describe the signals in each interface.

Avalon-ST TX Interface

Table 5–6 lists the signals in the Avalon-ST TX interface.

Table 5–6. Avalon-ST TX Signals

Signal Name	Direction	Description
<code>tx_parallel_data<n>[63:0]</code>	Sink	Avalon-ST data bus driven from the FPGA fabric to the TX PCS.
<code>tx_parallel_data<n>[64]</code>	Sink	Indicates whether <code>tx_parallel_data<n>[63:0]</code> represents control or data. When deasserted, <code>tx_parallel_data<n>[63:0]</code> is a data word. When asserted, <code>tx_parallel_data<n>[63:0]</code> is a control word. The value of header synchronization bits[65:64] of the Interlaken word identify whether bits[63:0] are a Framing Layer Control/Burst/IDLE Control Word or a data word. The MAC must gray encode the header synchronization bits. The value 2'b10 indicating Burst/IDLE Control Word must be gray encoded to the value 1'b1 for <code>tx_parallel_data<n>[64]</code> . The value 2'b01 indicating data word must be gray encoded to the value 1'b0 for <code>tx_parallel_data<n>[64]</code> .
<code>tx_parallel_data<n>[65]</code>	Sink	When asserted, indicates that <code>tx_parallel_data<n>[63:0]</code> is valid and is ready to be written into the TX FIFO. When deasserted, indicates that <code>tx_parallel_data<n>[63:0]</code> is invalid and is not written into the TX FIFO. The Interlaken MAC should gate <code>tx_parallel_data<n>[65]</code> based on <code>tx_datain_bp<n></code> .
<code>tx_ready</code>	Source	When asserted, indicates that the TX interface has exited the reset state and is ready for service. The <code>tx_ready</code> latency for the TX interface is 0. A 0 latency means that the TX FIFO can accept data on the same clock cycle that <code>tx_ready</code> is asserted if <code>tx_parallel_data[65]</code> (valid) is also asserted. The Interlaken MAC must wait for <code>tx_parallel_data[65]</code> before initiating data transfer on any lanes. The TX FIFO only captures input data from the Interlaken MAC when <code>tx_ready</code> and <code>tx_parallel_data[65]</code> are both asserted. For more information about the Avalon-ST interface and the use of ready signals, refer to the “Data Transfer with Backpressure” section in the “Avalon Streaming Interfaces” chapter of the <i>Avalon Interface Specifications</i> .
<code>tx_datain_bp<n></code>	Source	When asserted, indicates that Interlaken TX lane <code><n></code> interface is ready to receive data for transmission. The latency on this Avalon-ST interface is 0 cycles. The Interlaken MAC should drive <code>tx_parallel_data<n>[63:0]</code> as soon as <code>tx_ready<n></code> and <code>tx_sync_done</code> are asserted. The <code>tx_datain_bp<n></code> signal is connected to the <code>~partialfull</code> of the TX FIFO, so that when <code>tx_datain_bp<n></code> is deasserted the TX FIFO is almost full and back pressures the Interlaken MAC.

Table 5–6. Avalon-ST TX Signals

Signal Name	Direction	Description
tx_clkout	Output	Output clock from the TX PCS. The frequency of this clock equals the Lane rate divided by 40, which is the PMA serialization factor.
tx_user_clkout	Output	Master channel tx_clkout is available when you do not create the optional tx_coreclkln.
pll_locked	Output	In multilane Interlaken designs, this signal is the bitwise AND of the individual lane pll_locked signals.
tx_sync_done	Output	When asserted, indicates that all tx_parallel_data lanes are synchronized and ready for service. The Interlaken MAC must wait for this signal and tx_ready before initiating data transfer on any lane.

Avalon-ST RX Interface

Table 5–7 describes the signals in the Avalon-ST RX interface.

Table 5–7. Avalon-ST RX Signals (Part 1 of 3)

Signal Name	Direction	Description
rx_parallel_data<n>[63:0]	Source	Avalon-ST data bus driven from the RX PCS to the FPGA fabric.
rx_parallel_data<n>[64]	Source	When asserted, indicates that rx_parallel_data<n>[63:0] is valid. When deasserted, indicates the rx_parallel_data<n>[63:0] is invalid. The Interlaken PCS implements a gearbox between the PMA and PCS interface. The rx_parallel_data<n>[64] port is deasserted whenever the gearbox is in the invalid region. The Interlaken MAC should gate rx_dataout_bp<n> usage based on this signal.
rx_parallel_data<n>[65]	Source	Indicates whether rx_parallel_data<n>[63:0] represents control or data. When deasserted, rx_parallel_data<n>[63:0] is a data word. When asserted, rx_parallel_data<n>[63:0] is a control word. The value of header synchronization bits[65:64] of the Interlaken word identify whether bits[63:0] are Framing Layer Control/Burst/IDLE Word or a data word. The value 2'b10 indicating a Framing Layer Control/Burst/IDLE Word is gray encoded to the value 1'b1 and rx_parallel_data<n>[65] is asserted by the Interlaken Receive PCS. The value 2'b01 indicating data word is gray encoded to the value 1'b0 and rx_parallel_data<n>[65] is deasserted by the Interlaken Receive PCS. The Framing Layer Control Words are not discarded and are sent to the Interlaken MAC for multi-lane alignment and deskew on the lanes.

Table 5-7. Avalon-ST RX Signals (Part 2 of 3)

Signal Name	Direction	Description
<code>rx_parallel_data<n>[66]</code>	Source	This is an active-high synchronous status signal indicating that block lock (frame synchronization) and frame lock (metaframe boundary delineation) have been achieved. The Interlaken MAC should use this signal to indicate that Metaframe synchronization has been achieved for this lane. If the RX PCS FIFO reaches the empty state or is in an empty state, <code>rx_parallel_data<n>[66]</code> Block Lock and Frame Lock status signals are deasserted in the next clock cycle. <code>rx_parallel_data<n>[70]</code> indicating metaframe lock and <code>rx_parallel_data<n>[69]</code> indicating that the first Interlaken synchronization word alignment pattern has been received remain asserted.
<code>rx_parallel_data<n>[67]</code>	Source	When asserted, indicates an RX FIFO overflow error.
<code>rx_parallel_data<n>[68]</code>	Source	When asserted, indicates that the RX FIFO is partially empty and is still accepting data from the frame synchronizer. This signal is asserted when the RX FIFO has limited data. The Interlaken MAC should begin reading from the RX FIFO when this signal is deasserted, indicating sufficient FIFO contents to prevent underflow. The MAC should continue to read the RX FIFO to prevent overflow as long as this signal is not reasserted.
<code>rx_parallel_data<n>[69]</code>	Source	When asserted, indicates that the RX FIFO has found the first Interlaken synchronization word alignment pattern. For very short metaframes, this signal may be asserted after the frame synchronizer state machine validates frame synchronization and asserts <code>rx_parallel_data<n>[70]</code> because this signal is asserted by the RX FIFO which is the last PCS block in the RX datapath. This signal is optional. If the RX PCS FIFO reaches the empty state or is in an empty state, <code>rx_parallel_data<n>[70]</code> indicating metaframe lock and <code>rx_parallel_data<n>[69]</code> indicating that the first Interlaken synchronization word alignment pattern has been received remain asserted, but <code>rx_parallel_data<n>[66]</code> block lock and frame lock status signal are deasserted in the next clock cycle.
<code>rx_parallel_data<n>[70]</code>	Source	When asserted, indicates that the RX frame synchronization state machine has found and received 4 consecutive, valid synchronization words. The frame synchronization state machine requires 4 consecutive synchronization words to exit the presync state and enter the synchronized state. This signal is optional. If the RX PCS FIFO reaches an empty state or is in an empty state, <code>rx_parallel_data<n>[70]</code> indicating metaframe lock and <code>rx_parallel_data<n>[69]</code> indicating that the first Interlaken synchronization word alignment pattern has been received remain asserted but <code>rx_parallel_data<n>[66]</code> block lock and frame lock status signal are deasserted in the next clock cycle.
<code>rx_parallel_data<n>[71]</code>	Source	When asserted, indicates a CRC32 error in this lane. This signal is optional.
<code>rx_ready</code>	Source	When asserted, indicates that the RX interface has exited the reset state and is ready for service. The Interlaken MAC must wait for <code>rx_ready</code> to be asserted before initiating data transfer on any lanes.

Table 5-7. Avalon-ST RX Signals (Part 3 of 3)

Signal Name	Direction	Description
rx_clkout	Output	Output clock from the RX PCS. The frequency of this clock equals the Lane rate divided by 40, which is the PMA serialization factor.
rx_fifo_clr<n>	Input	When asserted, the RX FIFO is flushed. This signal allows you to clear the FIFO if synchronization is not achieved.
rx_dataout_bp<n>	Sink	When asserted, enables reading of data from the RX FIFO. This signal functions as a read enable. The RX interface has a ready latency of 1 cycle so that rx_paralleldata<n>[63:0] and rx_paralleldata<n>[65] are valid the cycle after rx_dataout_bp<n> is asserted.
rx_user_clkout	Output	Master channel rx_user_clkout is available when you do not create the optional rx_coreclkin.

PLL Interface

Table 5-9 describes the signals in the PLL interface.

Table 5-8. Serial Interface

Signal Name	Direction	Description
pll_ref_clk	Input	Reference clock for the PHY PLLs. Refer to the Lane rate entry in Table 5-2 on page 5-2 for required frequencies.

TX and RX Serial Interface

Table 5-9 describes the signals in the chip-to-chip serial interface.

Table 5-9. Serial Interface

Signal Name	Direction	Description
tx_serial_data	Output	Differential high speed serial output data using the PCML I/O standard. Clock is embedded in the serial data stream.
rx_serial_data	Input	Differential high speed serial input data using the PCML I/O standard. Clock is recovered from the serial data stream.

Optional Clocks for Deskew

Table 5-10 describes the optional clocks that you can create to reduce clock skew.

Table 5-10. Serial Interface

Signal Name	Direction	Description
tx_coreclkin	Input	When enabled tx_coreclkin is available as input port which drives the write side of TX FIFO. Altera recommends using this clock to reduce clock skew. When disabled, tx_clkout drives the write side the TX FIFO. tx_clkout must be used when the number of lanes is greater than 1.
rx_coreclkin	Input	When enabled, rx_coreclkin is available as input port which drives the read side of RX FIFO. Altera recommends using this clock to reduce clock skew. When disabled, rx_user_clkout, which is the master rx_clkout for all the bonded receiver lanes, is internally routed to drive the read side the RX FIFO.

Registers

The Avalon-MM PHY management interface provides access to the Interlaken PCS and PMA registers, resets, error handling, and serial loopback controls. You can use an embedded controller acting as an Avalon-MM master to send read and write commands to this Avalon-MM slave interface. Table 5-11 describes the signals that comprise the Avalon-MM management interface.

Table 5-11. Avalon-MM PCS Management Interface

Signal Name	Direction	Description
phy_mgmt_clk	Input	Avalon-MM clock input. There is no frequency restriction for Stratix V devices; however, if you plan to use the same clock for the PHY management interface and transceiver reconfiguration, you must restrict the frequency range of phy_mgmt_clk to 100–125 MHz to meet the specification for the transceiver reconfiguration clock.
phy_mgmt_clk_reset	Input	Global reset signal that resets the entire Interlaken PHY. Changed definition of phy_mgmt_clk_reset. This signal is active high and level sensitive. .
phy_mgmt_addr[8:0]	Input	9-bit Avalon-MM address.
phy_mgmt_writedata[31:0]	Input	Input data.
phy_mgmt_readdata[31:0]	Output	Output data.
phy_mgmt_write	Input	Write signal.
phy_mgmt_read	Input	Read signal.
phy_mgmt_waitrequest	Output	When asserted, indicates that the Avalon-MM slave interface is unable to respond to a read or write request. When asserted, control signals to the Avalon-MM slave interface must remain constant.

Register Descriptions

Table 5-12 specifies the registers that you can access using the Avalon-MM PHY management interface using word addresses and a 32-bit embedded processor. A single address space provides access to all registers. Writing to reserved or undefined register addresses may have undefined side effects.



All undefined register bits are reserved.

Table 5-12. Interlaken PHY Registers (Part 1 of 3)

Word Addr	Bits	R/W	Register Name	Description
PMA Common Control and Status Registers				
0x022	[31:0]	RO	pma_tx_pll_is_locked	Bit[P] indicates that the TX CMU PLL (P) is locked to the input reference clock. There is typically one pma_tx_pll_is_locked bit per system.
Reset Control Registers—Automatic Reset Controller				
0x041	[31:0]	RW	reset_ch_bitmask	Reset controller channel bitmask for digital resets. The default value is all 1s. Channel <n> can be reset when bit<n> = 1. Channel <n> cannot be reset when bit<n> = 0.

Table 5–12. Interlaken PHY Registers (Part 2 of 3)

Word Addr	Bits	R/W	Register Name	Description
0x042	[1:0]	WO	reset_control (write)	Writing a 1 to bit 0 initiates a TX digital reset using the reset controller module. The reset affects channels enabled in the reset_ch_bitmask. Writing a 1 to bit 1 initiates a RX digital reset of channels enabled in the reset_ch_bitmask.
		R	reset_status(read)	Reading bit 0 returns the status of the reset controller TX ready bit. Reading bit 1 returns the status of the reset controller RX ready bit.
Reset Controls –Manual Mode				
0x044	—	RW	reset_fine_control	You can use the reset_fine_control register to create your own reset sequence. The reset control module, illustrated in Figure 1–1 on page 1–2, performs a standard reset sequence at power on and whenever the phy_mgmt_clk_reset is asserted. Bits [31:4, 0] are reserved.
	[3]	RW	reset_rx_digital	Writing a 1 causes the RX digital reset signal to be asserted, resetting the RX digital channels enabled in reset_ch_bitmask. You must write a 0 to clear the reset condition.
	[2]	RW	reset_rx_analog	Writing a 1 causes the internal RX digital reset signal to be asserted, resetting the RX analog logic of all channels enabled in reset_ch_bitmask. You must write a 0 to clear the reset condition.
	[1]	RW	reset_tx_digital	Writing a 1 causes the internal TX digital reset signal to be asserted, resetting all channels enabled in reset_ch_bitmask. You must write a 0 to clear the reset condition.
PMA Control and Status Registers				
0x061	[31:0]	RW	phy_serial_loopback	Writing a 1 to channel <n> puts channel <n> in serial loopback mode. For information about pre- or post-CDRserial loopback modes, refer to “Loopback Modes” on page 10–39 .
0x064	[31:0]	RW	pma_rx_set_locktodata	When set, programs the RX CDR PLL to lock to the incoming data. Bit <n> corresponds to channel <n>.
0x065	[31:0]	RW	pma_rx_set_locktoref	When set, programs the RX CDR PLL to lock to the reference clock. Bit <n> corresponds to channel <n>.
0x066	[31:0]	R	pma_rx_is_lockedtodata	When asserted, indicates that the RX CDR PLL is locked to the RX data, and that the RX CDR has changed from LTR to LTD mode. Bit <n> corresponds to channel <n>.
00x067	[31:0]	R	pma_rx_is_lockedtoref	When asserted, indicates that the RX CDR PLL is locked to the reference clock. Bit <n> corresponds to channel <n>.
0x080	[31:0]	RW	indirect_addr	Provides for indirect addressing of all PCS control and status registers. Use this register to specify the logical channel address of the PCS channel you want to access.

Table 5-12. Interlaken PHY Registers (Part 3 of 3)

Word Addr	Bits	R/W	Register Name	Description
Stratix V Device Registers				
0x081	[27]	R	rx_crc32_err	Asserted by the CRC32 checker to indicate a CRC error in the corresponding RX lane. From block: CRC32 checker.
	[25]	R	rx_sync_lock	Asserted by the frame synchronizer to indicate that 4 frame synchronization words have been received so that the RX lane is synchronized. From block: Frame synchronizer.
	[24]	R	rx_word_lock	Asserted when the first alignment pattern is found. The RX FIFO generates this synchronous signal. From block: The RX FIFO generates this synchronous signal.

Transceiver Reconfiguration

As silicon progresses towards smaller process nodes, circuit performance is affected more by variations due to process, voltage, and temperature (PVT). These process variations result in analog voltages that can be offset from required ranges. The calibration performed by the dynamic reconfiguration interface compensates for variations due to PVT.

For Stratix V devices, each channel and each TX PLL have separate dynamics reconfiguration interfaces. The parameter editor provides informational messages on the connectivity of these interfaces. [Example 5-1](#) shows the messages for a 4-channel Interlaken PHY IP core.

Example 5-1. Informational Messages for the Transceiver Reconfiguration Interface

```
PHY IP will require 5 reconfiguration interfaces for connection to the external
reconfiguration controller.
Reconfiguration interface offsets 0-3 are connected to the transceiver channels.
Reconfiguration interface offset 4 is connected to the transmit PLL.
```

Although you must initially create a separate reconfiguration interface for each channel and TX PLL in your design, when the Quartus II software compiles your design, it reduces the number of reconfiguration interfaces by merging reconfiguration interfaces. The synthesized design typically includes a reconfiguration interface for three channels. Allowing the Quartus II software to merge reconfiguration interfaces gives the Fitter more flexibility in placing transceiver channels. For more information about transceiver reconfiguration refer to [Chapter 10, Transceiver Reconfiguration Controller](#).

Table 5–13 describes the signals in the reconfiguration interface. This interface uses the Avalon-MM PHY Management interface clock.

Table 5–13. Reconfiguration Interface

Signal Name	Direction	Description
reconfig_to_xcvr [(<i>n</i> >70)-1:0]	Sink	Reconfiguration signals from the Transceiver Reconfiguration Controller. <i>n</i> grows linearly with the number of reconfiguration interfaces. <i>n</i> initially includes the total number transceiver channels and TX PLLs before optimization/merging.
reconfig_from_xcvr [(<i>n</i> >46)-1:0]	Source	Reconfiguration signals to the Transceiver Reconfiguration Controller. <i>n</i> grows linearly with the number of reconfiguration interfaces. <i>n</i> initially includes the total number transceiver channels before optimization/merging.

TimeQuest Timing Constraints

You must add the following TimeQuest constraint to your Synopsys Design Constraints File (.sdc) timing constraint file:

```
derive_pll_clocks -create_base_clocks
```

Simulation Files and Example Testbench

Refer to “Running a Simulation Testbench” on page 1–4 for a description of the directories and files that the Quartus II software creates automatically when you generate your Interlaken PHY IP core.

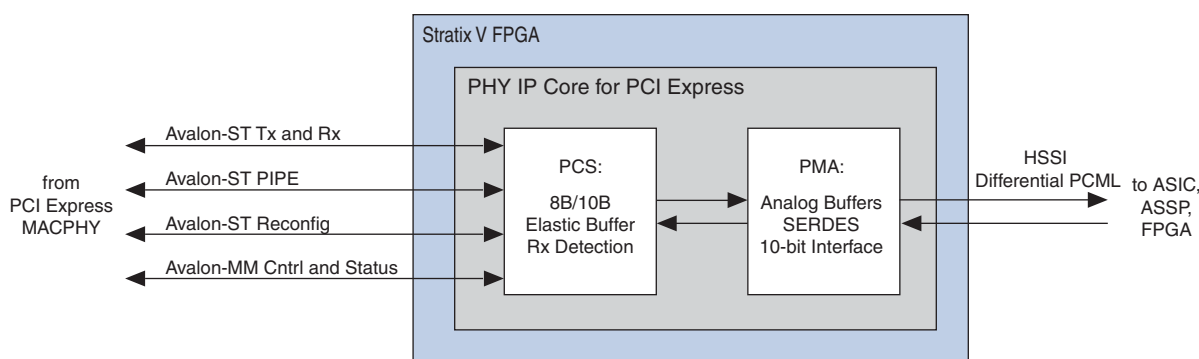


Refer to the [Altera wiki](#) for an example testbench that you can use as a starting point in creating your own verification environment.

The Altera PHY IP core for PCI Express (PIPE) implements physical coding sublayer (PCS) and physical media attachment (PMA) modules as defined by the *Intel PHY Interface for PCI Express (PIPE) Architecture* specification. The PHY IP core for PCI Express connects to a PHYMAC for PCI Express to create a complete design. Altera supports the Gen1 and Gen2 specifications and $\times 1$, $\times 4$, or $\times 8$ operation for a total aggregate bandwidth of 2 to 32 Gbps.

Figure 6–1 illustrates the top-level blocks of the PCI Express PHY (PIPE) for Stratix V GX devices.

Figure 6–1. PCI Express PHY (PIPE) with Hard IP PCS and PMA in Stratix V GX Devices



For more detailed information about the PCI Express PHY PIPE transceiver channel datapath, clocking, and channel placement, refer to the “PCI Express” section in the *Transceiver Protocol Configurations in Stratix V Devices* chapter of the *Stratix V Device Handbook*.

Device Family Support

IP cores provide either final or preliminary support for target Altera device families. These terms have the following definitions:

- *Final support*—Verified with final timing models for this device.
- *Preliminary support*—Verified with preliminary timing models for this device.

Table 6–1 shows the level of support offered by the PCI Express PIPE IP core for Altera device families

Table 6–1. Device Family Support

Device Family	Support
Stratix V devices—Hard PCS + PMA	Preliminary
Other device families	No support

Resource Utilization

Table 6–2 shows the typical expected device resource utilization for different configurations using the current version of the Quartus II software targeting a Stratix V GX device.

Table 6–2. PCI Express PHY (PIPE) Performance and Resource Utilization—Stratix V Devices

Number of Lanes	Combinational ALUTs	Logic Registers	Memory Bits	PLLs
Gen1 ×1	460	285	0	2
Gen1 ×4	530	373	0	5
Gen1 ×8	590	425	0	9
Gen2 ×1	460	295	0	2
Gen2 ×4	530	373	0	5
Gen2 ×8	590	425	0	9

Parameter Settings

To configure the PHY IP core for PCI Express in the parameter editor, click **Installed Plug-Ins > Interfaces > PCI Express > PHY IP Core for PCI Express (PIPE) v11.1**. This PHY IP core is only available when you select the **Stratix V** device family.

General Options

This section describes the PHY IP core for PCI Express parameters, which you can set using the parameter editor. Table 6–3 lists the settings available on **General Options** tab.

Table 6–3. General Options (Part 1 of 2)

Name	Value	Description
Device family	Stratix V	Supports only Stratix V devices
Number of lanes	1, 4, 8	The total number of duplex lanes
Protocol version	Gen1 (2.5 Gbps) Gen2 (5.0 Gbps)	Specifies the protocol version. Gen1 implements <i>PCI Express Base Specification 1.1</i> . Gen2 implements <i>PCI Express Base Specification 2.0</i> .
Base data rate	1 × Lane rate 2 × Lane rate 4 × Lane rate 8 × Lane rate	The base data rate is the output clock frequency of the PLL. Select a base data rate that minimizes the number of PLLs required to generate all the clock s required for data transmission. By selecting an appropriate base data rate , you can achieve the required data rate by changing the divider used by the clock generation block.

Table 6–3. General Options (Part 2 of 2)

Name	Value	Description
PLL type	CMU ATX	You can select either the CMU or ATX PLL. The CMU PLL has a larger frequency range than the ATX PLL. The ATX PLL is designed to improve jitter performance and achieves lower channel-to-channel skew; however, it supports a narrower range of data rates and reference clock frequencies. For example, if a base data rate of 2500 Mbps is not available with the ATX PLL and a 100 MHz reference clock; however, base data rates of 5000 Mbps or 10000 Mbps are possible with the ATX PLL and 100 MHz reference clock. Another advantage of the ATX PLL is that it does not use a transceiver channel, while the CMU PLL does.
PLL reference clock frequency	100 MHz 125 MHz	If you specify a CMU PLL for the TX PLL, you can use either the 100 MHz or 125 MHz input reference clock.
Deserialization factor	8, 16	Specifies the width of the interface between the PHYMAC and PHY (PIPE). Using the 16-bit interface, reduces the required clock frequency by half at the expense of extra FPGA resources.
PIPE low latency synchronous mode	On/Off	When enabled, the rate match FIFO is in low latency mode.
Run length	5–160	Specifies the legal number of consecutive 0s or 1s.
Enable electrical idle inferencing	True/False	When True , enables the PIPE interface to infer electrical idle instead of detecting electrical idle using analog circuitry. For more information about inferring electrical idle, refer to “ Section 4.2.3.4 Inferring Electrical Idle ” in the <i>PCI Express Base Specification 2.0</i> .

Analog Options

You specify the analog parameters for Stratix V devices using the Quartus II Assignment Editor, the Pin Planner, or through the Quartus II Settings File (.qsf). The default values for analog options fall into three categories:

- *Global*—These parameters have default values that are independent of other parameter settings.
- *Computed*—These parameters have an initial default value that is recomputed based on other parameter settings.
- *Proxy*—These parameters have default values that are place holders. The Quartus II software selects these initial default values based on your design; however, Altera recommends that you replace these defaults with values that match your electrical board specification.

Table 6-4 lists the analog parameters for Stratix V devices whose original values are place holders for the values that match your electrical board specification. In Table 6-4, the default value of an analog parameter is shown in **bold** type.

Table 6-4. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 2)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_IO_PIN_TERMINATION	Transceiver I/O Pin Termination	Specifies the intended on-chip termination value for the specified transceiver pin. Use External Resistor if you intend to use off-chip termination.	85_OHMS 100_OHMS 120_OHMS 150_OHMS EXTERNAL_RESISTOR	Pin
XCVR_REFCLK_PIN_TERMINATION	Transceiver Dedicated Refclk Pin Termination	Specifies the intended termination value for the specified refclk pin.	DC_COUPLING_INTERNAL_100_OHM DC_COUPLING_EXTERNAL_RESISTOR AC_COUPLING	Pin
XCVR_RX_BYPASS_EQ_STAGES_234	Receiver Equalizer Stage 2, 3, 4 Bypass	Bypass continuous time equalizer stages 2, 3, and 4 to save power. This setting eliminates significant AC gain on the equalizer and is appropriate for chip-to-chip short range communication on a PCB.	ALL_STAGES_ENABLED BYPASS_STAGES	Pin
XCVR_TX_SLEW_RATE_CTRL	Transmitter Slew Rate Control	Specifies the slew rate of the output signal. The valid values span from the slowest rate to fastest rate with 1 representing the slowest rate.	1-5	Pin

Table 6–4. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 2)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_VCCA_VOLTAGE	VCCA_GXB Voltage	Configure the VCCA_GXB voltage for a GXB I/O pin by specifying the intended VCCA_GXB voltage for a GXB I/O pin. If you do not make this assignment the compiler automatically sets the correct VCCA_GXB voltage depending on the configured data rate, as follows: ■ Data rate <= 6.5 Gbps: 2_5V ■ Data rate > 6.5 Gbps: 3_0V or 3_3V for Stratix V ES silicon	2_5V 3_0V	Pin
XCVR_VCCR_VCCT_VOLTAGE	VCCR_GXB VCCT_GXB Voltage	Configure the VCCR_GXB and VCCT_GXB voltage for an GXB I/O pin by specifying the intended supply voltages for a GXB I/O pin. If your design uses decision feedback equalization (DFE) or adaptive equalization (AEQ), you must set this parameter to 1.0V. Otherwise, if you do not make this assignment the compiler automatically sets the correct VCCR_GXB and VCCT_GXB voltage depending on the configured data rate as follows: ■ Data rate <= 6.5 Gbps: 0_85V ■ Data rate > 6.5 Gbps: 1_0V	0_85V 1_0V	Pin

Table 6–5 lists the analog parameters with *global* or *computed* default values. You may want to optimize some of these settings. In Table 6–5, the default value is shown in **bold** type. For computed analog parameters, the default value listed is for the initial setting, not the recomputed setting.

Table 6–5. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 3)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
Analog Parameters with Global Default Value				
CDR_BANDWIDTH_PRESET	CDR Bandwidth Preset	Specifies the CDR bandwidth preset setting.	Auto Low Medium High	PLL instance
PLL_BANDWIDTH_PRESET	PLL Bandwidth Preset	Specifies the PLL bandwidth preset setting	Auto Low Medium High	PLL instance
XCVR_RX_DC_GAIN	Receiver Buffer DC Gain Control	Controls the amount of a stage receive-buffer DC gain.	0–4	Pin

Table 6–5. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 3)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_RX_LINEAR_EQUALIZER_CONTROL	Receiver Linear Equalizer Control	Static control for the continuous time equalizer in the receiver buffer. The equalizer has 16 distinct settings from 0–15 corresponding to the increasing AC gain.	1–16	Pin
Analog Parameter with Computed Default Value				
XCVR_RX_COMMON_MODE_VOLTAGE	Receiver Buffer Common Mode Voltage	Receiver buffer common-mode voltage.	VTT_OP80V VTT_OP75V VTT_OP70V VTT_OP65V VTT_OP60V VTT_OP55V VTT_OP50V VTT_OP35V VTT_PUP _WEAK VTT_PDN WEAK TRISTATE1 VTT_PDN_ STRONG VTT_PUP_ STRONG TRISTATE2 TRISTATE3 TRISTATE4	Pin
XCVR_RX_ENABLE_LINEAR_EQUALIZER_PCIEMODE	Receiver Linear Equalizer Control (PCI Express)	If enabled equalizer gain control is driven by the PCS block for PCI Express. If disabled equalizer gain control is determined by the XCVR_RX_LINEAR_EQUALIZER_SETTING assignment.	TRUE FALSE	Pin
XCVR_RX_EQ_BW_SEL	Receiver Equalizer Gain Bandwidth Select	Sets the gain peaking frequency for the equalizer. For data-rates of less than 6.5Gbps set to HALF. For higher data-rates set to FULL.	FULL HALF	Pin
XCVR_RX_SD_ENABLE	Receiver Signal Detection Unit Enable/Disable	Enables or disables the receiver signal detection unit.	TRUE FALSE	Pin
XCVR_RX_SD_OFF	Receiver Cycle Count Before Signal Detect Block Declares Loss Of Signal	Number of parallel cycles to wait before the signal detect block declares loss of signal.	0 –29	Pin
XCVR_RX_SD_ON	Receiver Cycle Count Before Signal Detect Block Declares Presence Of Signal	Number of parallel cycles to wait before the signal detect block declares presence of signal.	0 –16	Pin
XCVR_RX_SD_THRESHOLD	Receiver Signal Detection Voltage Threshold	Specifies signal detection voltage threshold level.	0 –7	Pin

Table 6–5. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 3)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_TX_COMMON_MODE_VOLTAGE	Transmitter Common Mode Driver Voltage	Transmitter common-mode driver voltage	VOLT_OP80V VOLT_OP75V VOLT_OP70V VOLT_OP65V VOLT_OP60V VOLT_OP55V VOLT_OP50V VOLT_OP35V PULL_UP PULL_DOWN TRISTATED1 GROUNDED PULL_UP_TO_VCCELA TRISTATED2 TRISTATED3 TRISTATED4	Pin
XCVR_TX_PRE_EMP_1ST_POST_TAP	Transmitter Preemphasis First Post-Tap	Specifies the first post-tap setting value.	0 –31	Pin
XCVR_TX_PRE_EMP_2ND_POST_TAP	Transmitter Preemphasis Second Post-Tap	Specifies the second post-tap setting value.	0 –15	Pin
XCVR_TX_PRE_EMP_INV_2ND_TAP	Transmitter Preemphasis Second Tap Invert	Inverts the transmitter pre-emphasis 2nd post tap.	TRUE FALSE	Pin
XCVR_TX_PRE_EMP_INV_PRE_TAP	Transmitter Preemphasis Pre Tap Invert	Inverts the transmitter pre-emphasis pre-tap.	TRUE FALSE	Pin
XCVR_TX_PRE_EMP_PRE_TAP	Transmitter Preemphasis Pre-Tap	Specifies the pre-tap pre-emphasis setting.	0 –15	Pin
XCVR_TX_RX_DET_ENABLE	Transmitter's Receiver Detect Block Enable	Enables or disables the receiver detector circuit at the transmitter.	TRUE FALSE	Pin
XCVR_TX_RX_DET_MODE	Transmitter's Receiver Detect Block Mode	Sets the mode for receiver detect block	0 –15	Pin
XCVR_TX_RX_DET_OUTPUT_SEL	Transmitter's Receiver Detect Block QPI/PCI Express Control	Determines QPI or PCI Express mode for the Receiver Detect block.	RX_DET_QPI_OUT RX_DET_PCIE_OUT	Pin
XCVR_TX_VOD	Transmitter Differential Output Voltage	Differential output voltage setting. The values are monotonically increasing with the driver main tap current strength.	0–63 50	Pin
XCVR_TX_VOD_PRE_EMP_CTRL_SRC	Transmitter V _{OD} /Preemphasis Control Source	When set to DYNAMIC_CTL, the PCS block controls the V _{OD} and preemphasis coefficients for PCI Express. When this assignment is set to RAM_CTL the V _{OD} and preemphasis are controlled by other assignments, such as XCVR_TX_PRE_EMP_1ST_POST_TAP.	DYNAMIC_CTL RAM_CTL	Pin

For more information about the Pin Planner, refer to [About the Pin Planner](#) in Quartus II Help. For more information about the Assignment Editor, refer to [About the Assignment Editor](#) in Quartus II Help.

For more information about Quartus II Settings, refer to [Quartus II Settings File Manual](#).

Interfaces

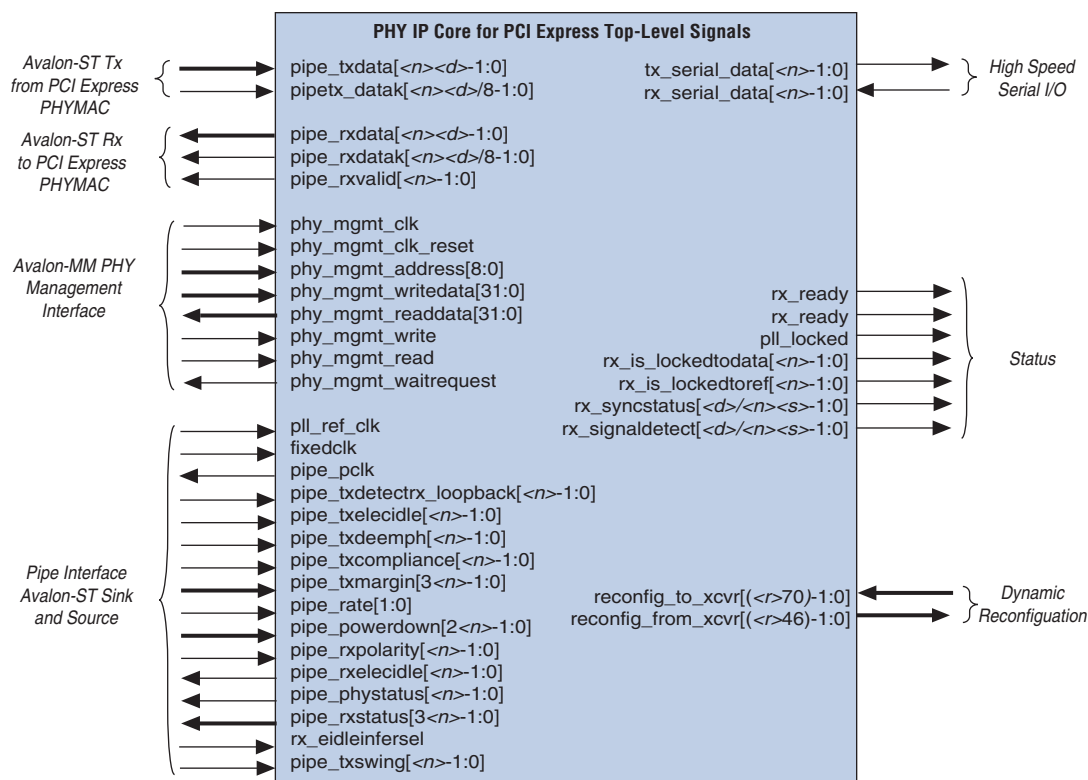
This section describes interfaces of the PHY IP Core for PCI Express (PIPE). It includes the following topics:

- Ports
- Registers
- Dynamic Reconfiguration

Ports

Figure 6-2 illustrates the top-level pinout of the PHY IP core for PCI Express PHY.

Figure 6-2. Top-Level Signals of the PHY IP Core for PCI Express⁽¹⁾



Note to Figure 6-2:

(1) `<n>` is the number of lanes. `<d>` is the total deserialization factor from the input pin to the PHYMAC interface. `<s>` is the symbols size. `<r>` is the width of the reconfiguration interface; `<r>` is automatically calculated based on the selected configuration.



The **block diagram** shown in the GUI labels the external pins with the *interface type* and places the *interface name* inside the box. The interface type and name are used in the `_hw.tcl` file. If you turn on **Show signals**, the **block diagram** displays all top-level signal names.



For more information about `_hw.tcl` files, refer to the *Component Interface Tcl Reference* chapter in volume 1 of the *Quartus II Handbook*.

The following sections describe the signals in each interface.

Avalon-ST TX Input Data from the PHYMAC

Table 6-6 describes the signals in the Avalon-ST input interface. These signals are driven from the PHYMAC to the PCS. This is an Avalon sink interface.



For more information about the Avalon-ST protocol, including timing diagrams, refer to the *Avalon Interface Specifications*.

Table 6-6. Avalon-ST TX Inputs

Signal Name	Dir	Description
<code>pipe_txdata[<n><d>-1:0]</code>	Sink	This is TX parallel data driven from the PHYMAC for PCI Express. The ready latency on this interface is 0, so that the PHY must be able to accept data as soon as the PHY exits reset.
<code>pipe_txdatak[<n><d>/8-1:0]</code>	Sink	Data and control indicator for the received data. When 0, indicates that <code>pipe_txdata</code> is data, when 1, indicates that <code>pipe_txdata</code> is control.

Avalon-ST RX Output Data to the PHYMAC

Table 6-7 describes the signals in the Avalon-ST output interface. These signals are driven from the PHY to the PHYMAC. This is an Avalon source interface.

Table 6-7. Avalon-ST RX Inputs

Signal Name	Dir	Description
<code>pipe_rxdata[<n><d>-1:0]</code>	Source	This is RX parallel data driven from the PHY. The ready latency on this interface is 0, so that the MAC must be able to accept data as soon as the PHY comes out of reset.
<code>pipe_rxdatak[<n><d>/8-1:0]</code>	Source	Data and control indicator for the source data. Bit 0 correspond the low byte of <code>pipe_rxdata</code> . Bit 1 corresponds to the upper byte. When 0, indicates that <code>pipe_rxdata</code> is data, when 1, indicates that <code>pipe_rxdata</code> is control.
<code>pipe_rxvalid[<n>-1:0]</code>	Source	Asserted when RX data and control are valid.

PIPE Interface

Table 6–8 describes the signals in the PIPE interface.

Table 6–8. PIPE Interface (Part 1 of 2)

Signal Name	Direction	Description
pll_ref_clk	Sink	This is the 100 MHz input reference clock source for the PHY PLL. You can optionally provide a 125 MHz input reference clock by setting the PLL reference clock frequency parameter to 125 MHz as described in Table 6–3 on page 6–2 . If you have enabled Configuration via Protocol (CvP) and your design includes other transceiver PHYs connected to the same Transceiver Reconfiguration Controller, then you should connect pll_ref_clk to the phy_mgmt_clk_clk signal of the Transceiver Reconfiguration Controller and the other transceiver PHYs. In addition, if your design includes more than one Transceiver Reconfiguration Controller on the same side of the FPGA, they all must share the phy_mgmt_clk_clk signal.
fixedclk	Sink	A 125 MHz clock used for the receiver detect circuitry. You must connect a 125 MHz input clock signal for the fixedclk port. This clock can be derived from pll_ref_clk.
pipe_txdetectrx_loopback	Sink	This signal instructs the PHY to start a receive detection operation. After power-up asserting this signal starts a loopback operation. Refer to section 6.4 of the <i>Intel PHY Interface for PCI Express (PIPE) Architecture</i> for a timing diagram.
pipe_txelecidle	Sink	This signal forces the transmit output to electrical idle. Refer to section 7.3 of the <i>Intel PHY Interface for PCI Express (PIPE) Architecture</i> for timing diagrams.
pipe_txdeemph	Sink	Transmit de-emphasis selection. In PCI Express Gen2 (5 Gbps) mode it selects the transmitter de-emphasis: ■ 1'b0: -6 dB ■ 1'b1: -3.5 dB
pipe_txcompliance	Sink	When asserted for one cycle, sets the 8B/10B encoder output running disparity to negative. Used when transmitting the compliance pattern. Refer to section 6.11 of the <i>Intel PHY Interface for PCI Express (PIPE) Architecture</i> for more information.
pipe_txmargin	Sink	Transmit V _{OD} margin selection. The PCI Express MegaCore® function hard IP sets the value for this signal based on the value from the Link Control 2 Register. This is 3 bits in the PIPE Specification.
pipe_rate	Sink	Specifies the link frequency, as follows: ■ 0 –Gen1 operation, or 2.5 Gbps ■ 1–Gen2 operation, or 5.0 Gbps Figure 6–3 on page 6–12 illustrates the timing of a rate switch from Gen1 to Gen2 and back to Gen1.

Table 6–8. PIPE Interface (Part 2 of 2)

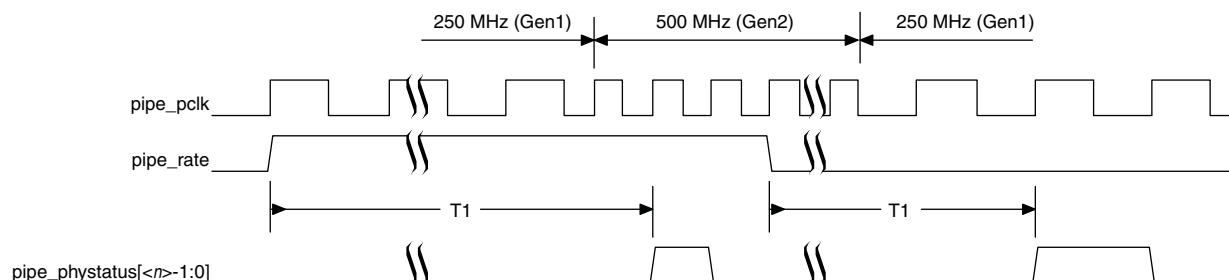
Signal Name	Direction	Description
pipe_powerdown<n>[1:0] ⁽¹⁾	Sink	This signal requests the PHY to change its power state to the specified state. The following encodings are defined: ■ 2b'00– P0, normal operation ■ 2b'01–P0s, low recovery time latency, power saving state ■ 2b'10–P1, longer recovery time (64 us maximum latency), lower power state ■ 2b'11–P2, lowest power state. (not supported)
pipe_rxpolarity	Sink	When 1, instructs the PHY layer to invert the polarity on the 8B/10B receiver decoding block.
pipe_rxelecidle	Source	When asserted, indicates receiver detection of an electrical idle.
pipe_phystatus	Source	This signal is used to communicate completion of several PHY requests.
pipe_rxstatus<n>[2:0] ⁽¹⁾	Source	This signal encodes receive status and error codes for the receive data stream and receiver detection. The following encodings are defined: ■ 000–receive data OK ■ 001–1 SKP added ■ 010–1 SKP removed ■ 011–Receiver detected ■ 100–Both 8B/10B decode error and (optionally) RX disparity error ■ 101–Elastic buffer overflow ■ 110–Elastic buffer underflow ■ 111–Receive disparity error.
rx_eidleinfersel[<n>-1:0]	Sink	When asserted high, the electrical idle state is inferred instead of being identified using analog circuitry to detect a device at the other end of the link. You can select electrical idle inferencing, by setting the Enable electrical idle inferencing parameter in the parameter editor to true .
pipe_txswing	Source	Indicates whether the transceiver is using full- or low-swing voltages as defined by the tx_pipemargin. ■ 0–Full swing ■ 1–Low swing

Note to Table 6–8:

(1) <n> is the number of lanes. The PHY (PIPE) supports ×1, ×4, ×8 operation.

Figure 6-3 illustrates the pipe_pclk switching from Gen1 to Gen2 and back to Gen1.

Figure 6-3. Rate Switch from Gen1 to Gen2



Note to Figure 6-3:

- (1) Time T1 is pending characterization.
- (2) $\langle n \rangle$ is the number of lanes.

Transceiver Serial Interface

Table 6-9 describes the differential serial TX and RX connections to FPGA pins.

Table 6-9. Transceiver Differential Serial Interface

Signal Name	Direction	Description
rx_serial_data[$\langle n \rangle$ -1:0]	Input	Receiver differential serial input data, $\langle n \rangle$ is the number of lanes.
tx_serial_data[$\langle n \rangle$ -1:0]	Output	Transmitter differential serial output data $\langle n \rangle$ is the number of lanes.

 For information about channel placement of the Hard IP PCI Express IP Core, refer to the *Channel Placement Gen1 and Gen2* and *Channel Placement Gen3* sections in the *Stratix V Hard IP for PCI Express User Guide*.

 For soft IP implementations of PCI Express, channel placement is determined by the Quartus II fitter.

Optional Status Interface

Table 6-10 describes the signals the optional status signals.

Table 6-10. Status Signals (Part 1 of 2)⁽¹⁾

Signal Name	Direction	Signal Name
tx_ready	Output	When asserted, indicates that the TX interface has exited the reset state and is ready to transmit.
rx_ready	Output	When asserted, indicates that the RX interface has exited the reset state and is ready to receive.
pll_locked[$\langle p \rangle$ -1:0]	Output	When asserted, indicates that the PLL is locked to the input reference clock. This signal is asynchronous.
rx_is_lockedtodata[$\langle n \rangle$ -1:0]	Output	When asserted, the receiver CDR is in to lock-to-data mode. When deasserted, the receiver CDR lock mode depends on the rx_locktorefclk signal level.
rx_is_lockedtoref[$\langle n \rangle$ -1:0]	Output	Asserted when the receiver CDR is locked to the input reference clock. This signal is asynchronous.

Table 6-10. Status Signals (Part 2 of 2) ⁽¹⁾

Signal Name	Direction	Signal Name
<code>rx_syncstatus[<d><n>/8-1:0]</code>	Output	Indicates presence or absence of synchronization on the RX interface. Asserted when word aligner identifies the word alignment pattern or synchronization code groups in the received data stream.
<code>rx_signaldetect[<d><n>/8-1:0]</code>	Output	When asserted indicates that the lane detects a sender at the other end of the link.

Note to Table 6-10:

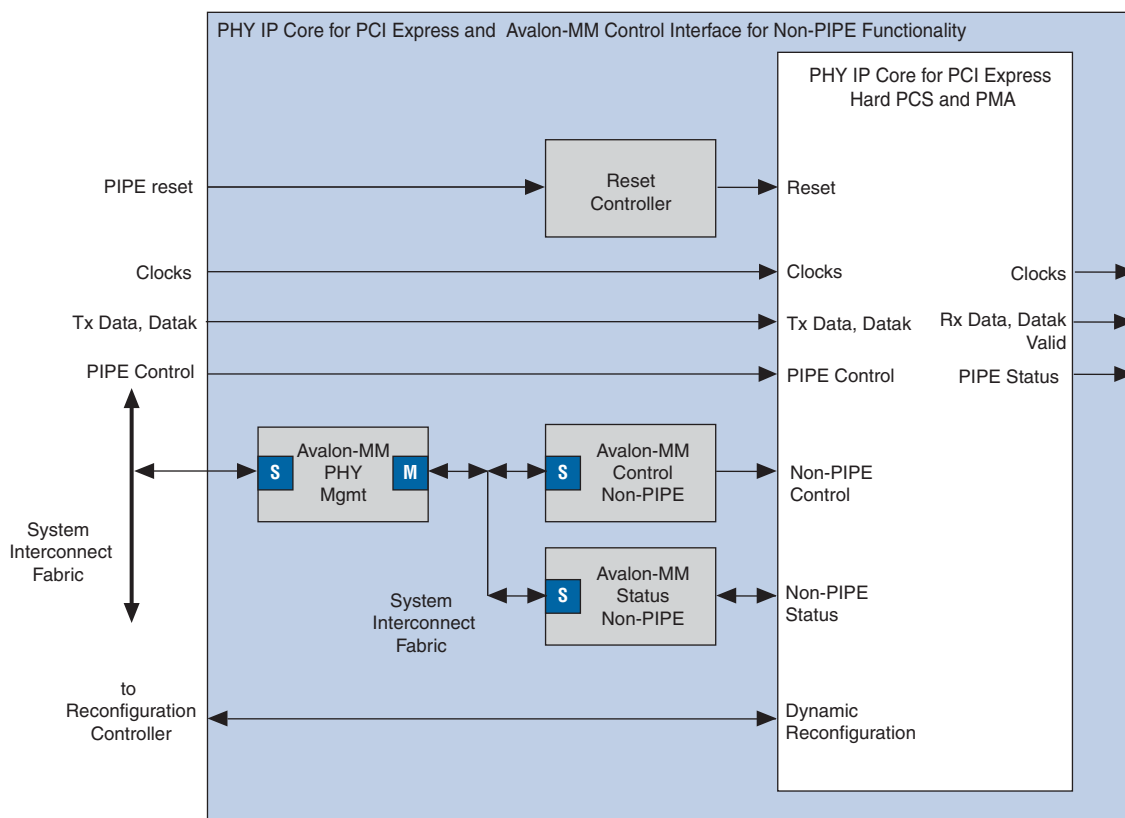
(1) <n> is the number of lanes. <d> is the deserialization factor. <p> is the number of PLLs.

Registers

The Avalon-MM PHY management interface provides access to the PHY IP Core for PCI Express PCS and PMA features that are not part of the standard PIPE interface. You can use an embedded controller acting as an Avalon-MM master to send read and write commands to this Avalon-MM slave interface.

Figure 6-4 provides a high-level view of this hardware.

Figure 6-4. PCI Express PIPE IP Core ⁽¹⁾



Note to Figure 6-4:

(1) Blocks in gray are soft logic. Blocks in white are hard logic.

Table 6–11 describes the signals that comprise the Avalon-MM PHY Management interface.

Table 6–11. Avalon-MM PHY Management Interface

Signal Name	Direction	Description
phy_mgmt_clk	Input	Avalon-MM clock input. There is no frequency restriction for Stratix V devices; however, if you plan to use the same clock for the PHY management interface and transceiver reconfiguration, you must restrict the frequency range of phy_mgmt_clk to 100–125 MHz to meet the specification for the transceiver reconfiguration clock.
phy_mgmt_clk_reset	Input	Global reset signal that resets the entire PHY IP core. Changed definition of phy_mgmt_clk_reset. This signal is active high and level sensitive.
phy_mgmt_address[8:0]	Input	9-bit Avalon-MM address.
phy_mgmt_writedata[31:0]	Input	Input data.
phy_mgmt_readdata[31:0]	Output	Output data.
phy_mgmt_write	Input	Write signal.
phy_mgmt_read	Input	Read signal.
phy_mgmt_waitrequest	Output	When asserted, indicates that the Avalon-MM slave interface is unable to respond to a read or write request. When asserted, control signals to the Avalon-MM slave interface must remain constant.

Register Descriptions

Table 6–12 describes the registers that you can access over the Avalon-MM PHY management interface using word addresses and a 32-bit embedded processor. A single address space provides access to all registers.



Writing to reserved or undefined register addresses may have undefined side effects.

Table 6–12. PCI Express PHY (PIPE) IP Core Registers (Part 1 of 4)

Word Addr	Bits	R/W	Register Name	Description
PMA Common Control and Status Registers				
0x022	[31:0]	R	pma_tx_pll_is_locked	Bit[P] indicates that the TX CMU PLL (P) is locked to the input reference clock. There is typically one pma_tx_pll_is_locked bit per system.
Reset Control Registers—Automatic Reset Controller				
0x041	[31:0]	RW	reset_ch_bitmask	Reset controller channel bitmask for digital resets. The default value is all 1s. Channel <n> can be reset when bit<n> = 1.

Table 6–12. PCI Express PHY (PIPE) IP Core Registers (Part 2 of 4)

Word Addr	Bits	R/W	Register Name	Description
0x042	[1:0]	W	reset_control (write)	Writing a 1 to bit 0 initiates a TX digital reset using the reset controller module. The reset affects channels enabled in the reset_ch_bitmask. Writing a 1 to bit 1 initiates a RX digital reset of channels enabled in the reset_ch_bitmask.
		R	reset_status(read)	Reading bit 0 returns the status of the reset controller TX ready bit. Reading bit 1 returns the status of the reset controller RX ready bit.
Reset Controls –Manual Mode				
0x044	[31:0]	RW	reset_fine_control	You can use the reset_fine_control register to create your own reset sequence. The reset control module, illustrated in Figure 1–1 on page 1–2 , performs a standard reset sequence at power on and whenever the phy_mgmt_clk_reset is asserted. Bits [31:4, 0] are reserved.
	[31:4]	RW	Reserved	It is safe to write 0s to reserved bits.
	[3]	RW	reset_rx_digital	Writing a 1 causes the RX digital reset signal to be asserted, resetting the RX digital channels enabled in reset_ch_bitmask. You must write a 0 to clear the reset condition.
	[2]	RW	reset_rx_analog	Writing a 1 causes the internal RX digital reset signal to be asserted, resetting the RX analog logic of all channels enabled in reset_ch_bitmask. You must write a 0 to clear the reset condition.
	[1]	RW	reset_tx_digital	Writing a 1 causes the internal TX digital reset signal to be asserted, resetting all channels enabled in reset_ch_bitmask. You must write a 0 to clear the reset condition.
	[0]	RW	pll_powerdown	Writing a 1 causes the internal TX PLL to powerdown. If you reset the transceiver, you must assert pll_powerdown by writing a 1 to this register and then writing a 0 after 1 μs. Asserting pll_powerdown also asserts tx_analogreset which is required for Gen2 operation.
PMA Control and Status Registers				
0x061	[31:0]	RW	phy_serial_loopback	Writing a 1 to channel <n> puts channel <n> in serial loopback mode.
0x063	[31:0]	R	pma_rx_signaldetect	When channel <n> =1, indicates that receive circuit for channel <n> senses the specified voltage exists at the RX input buffer. This option is only operational for the PCI Express PHY IP core.
0x064	[31:0]	RW	pma_rx_set_locktodata	When set, programs the RX CDR PLL to lock to the incoming data. Bit <n> corresponds to channel <n>.
0x065	[31:0]	RW	pma_rx_set_locktoref	When set, programs the RX CDR PLL to lock to the reference clock. Bit <n> corresponds to channel <n>.

Table 6–12. PCI Express PHY (PIPE) IP Core Registers (Part 3 of 4)

Word Addr	Bits	R/W	Register Name	Description
0x066	[31:0]	R	pma_rx_is_lockedtodata	When asserted, indicates that the RX CDR PLL is locked to the RX data, and that the RX CDR has changed from LTR to LTD mode. Bit <n> corresponds to channel <n>.
00x067	[31:0]	R	pma_rx_is_lockedtoref	When asserted, indicates that the RX CDR PLL is locked to the reference clock. Bit <n> corresponds to channel <n>.
PCS for PCI Express				
0x080	[31:0]	RW	Lane or group number	Specifies lane or group number for indirect addressing, which is used for all PCS control and status registers. For variants that stripe data across multiple lanes, this is the logical group number. For non-bonded applications, this is the logical lane number.
0x081	[31:6]	R	Reserved	—
	[5:1]	R	rx_bitslipboundary_selectout	Records the number of bits slipped by the RX Word Aligner to achieve word alignment. Used for very latency sensitive protocols. From block: Word aligner.
	[0]	R	rx_phase_comp_fifo_error	When set, indicates an RX phase compensation FIFO error. From block: RX phase compensation FIFO.
0x082	[31:1]	R	Reserved	—
	[0]	RW	tx_phase_comp_fifo_error	When set, indicates a TX phase compensation FIFO error. From block: TX phase compensation FIFO.
0x083	[31:6]	RW	Reserved	—
	[5:1]	RW	tx_bitslipboundary_select	Records the number of bits slipped by the TX bit slipper in the TX serial output. Used for very latency sensitive protocols. From block: TX bit-slipper.
	[0]	RW	tx_invpolarity	When set, the TX channel inverts the polarity of the TX data. To block: Serializer.
0x084	[31:1]	RW	Reserved	—
	[0]	RW	rx_invpolarity	When set, the RX channel inverts the polarity of the received data. The 8B/10B decoder inverts the decoder input sample and then decodes the inverted samples. To block: 8B/10B decoder.

Table 6-12. PCI Express PHY (PIPE) IP Core Registers (Part 4 of 4)

Word Addr	Bits	R/W	Register Name	Description
0x085	[31:4]	RW	Reserved	—
	[3]	RW	rx_bitslip	When set, the word alignment logic operates in bitslip mode. Every time this register transitions from 0 to 1, the RX data slips a single bit. To block: Word aligner.
	[2]	RW	rx_bytereversal_enable	When set, enables byte reversal on the RX interface. To block: Word aligner.
	[1]	RW	rx_bitreversal_enable	When set, enables bit reversal on the RX interface. To block: Word aligner.
	[0]	RW	rx_enapatternalign	When set, the word alignment logic operates in pattern detect mode. To block: Word aligner.
0x086	[31:20]	R	Reserved	—
	[19:16]	R	rx_rlv	When set, indicates a run length violation. From block: Word aligner.
	[15:12]	R	rx_patterndetect	When set, indicates that RX word aligner has achieved synchronization. From block: Word aligner.
	[11:8]	R	rx_disperr	When set, indicates that the received 10-bit code or data group has a disparity error. When set, the corresponding errdetect bits are also set. From block: 8B/10B decoder.
	[7:4]	R	rx_syncstatus	When set, indicates that the RX interface is synchronized to the incoming data. From block: Word aligner.
	[3:0]	R	rx_errdetect	When set, indicates that a received 10-bit code group has an 8B/10B code violation or disparity error. It is used along with RX disparity to differentiate between a code violation error and a disparity error, or both. In PIPE mode, the PIPE specific output port called <code>pipe_rxstatus</code> encodes the errors. From block: 8B/10B decoder.

Dynamic Reconfiguration

As silicon progresses towards smaller process nodes, circuit performance is affected more by variations due to process, voltage, and temperature (PVT). These process variations result in voltages that can be offset from required ranges. The calibration performed by the dynamic reconfiguration interface compensates for variations due to PVT.

For Stratix V devices, each channel and each TX PLL have separate dynamic reconfiguration interfaces. The parameter editor provides informational messages on the connectivity of these interfaces. [Example 6-1](#) shows the messages for a 8-channel PHY IP core for PCI Express (PIPE).

Although you must initially create a separate reconfiguration interface for each channel and TX PLL in your design, when the Quartus II software compiles your design, it reduces the number of reconfiguration interfaces by merging reconfiguration interfaces. The synthesized design typically includes a reconfiguration interface for three channels. Allowing the Quartus II software to merge reconfiguration interfaces gives the Fitter more flexibility in placing transceiver channels.

Example 6-1. Informational Messages for the Transceiver Reconfiguration Interface

PHY IP will require 9 reconfiguration interfaces for connection to the external reconfiguration controller.
Reconfiguration interface offsets 0-7 are connected to the transceiver channels.
Reconfiguration interface offset 8 is connected to the transmit PLL.

[Table 6-13](#) describes the signals in the reconfiguration interface. This interface uses the Avalon-MM PHY Management interface clock.

Table 6-13. Reconfiguration Interface

Signal Name	Direction	Description
reconfig_to_xcvr [r 70-1:0]	Sink	Reconfiguration signals from the Transceiver Reconfiguration Controller. r grows linearly with the number of reconfiguration interfaces.
reconfig_from_xcvr [r 46-1:0]	Source	Reconfiguration signals to the Transceiver Reconfiguration Controller. r grows linearly with the number of reconfiguration interfaces.

Simulation Files and Example Testbench

Refer to “[Running a Simulation Testbench](#)” on [page 1-4](#) for a description of the directories and files that the Quartus II software creates automatically when you generate your PHY IP core for PCI Express.



Refer to the [Altera wiki](#) for an example testbench that you can use as a starting point in creating your own verification environment.

The Altera Custom PHY IP core is a generic PHY that you can customize for use in Arria V or Stratix V FPGAs. You can connect your application's MAC-layer logic to the Custom PHY to transmit and receive data at rates of 0.611–6.5536 Gbps for Arria V devices or 0.622–8.5 Gbps for Stratix V devices. You can parameterize the physical coding sublayer (PCS) to include the functions that your application requires. The following functions are available:

- 8B/10B encode and decode
- Three word alignment modes
- Rate matching
- Byte ordering

By setting the appropriate options using the parameter editor, you can configure the Custom PHY IP core to support many standard protocols, including all of the following protocols:

- Serial Data Converter (SDC(JESD204A))
- Serial digital interface (SDI)
- Ethernet (GbE)
- Serial RapidIO® (SRIO) 1.3
- Serial ATA (SATA) and sequential active serial (SAS) Gen1, Gen2, and Gen3
- Gigabit-capable passive optical network (GPON)

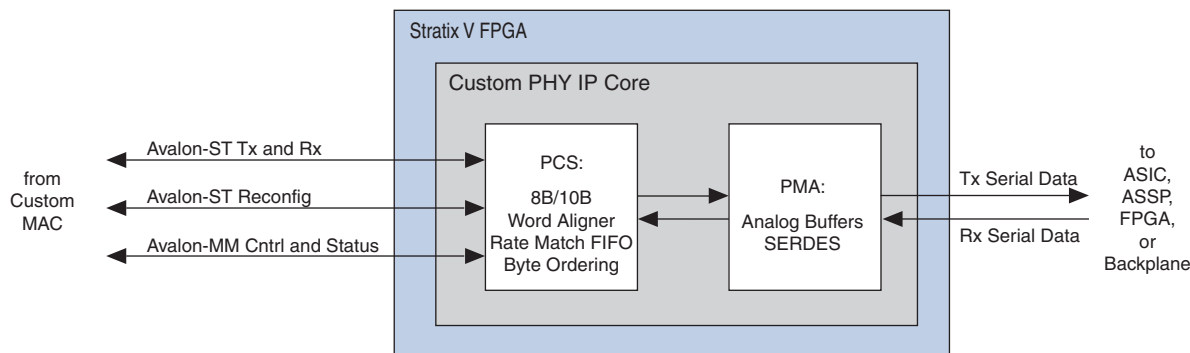
Your MAC layer must use the Avalon-ST to transmit and receive data from the Custom PHY. The Avalon-ST protocol is a simple protocol designed for driving high bandwidth, low latency, unidirectional data. To access control and status registers in the Custom PHY, your design must include an embedded controller with an Avalon-MM master interface. This is a standard, memory-mapped protocol that is typically used to read and write registers and memory.



For more information about the Avalon-ST and Avalon-MM protocols, refer to the *Avalon Interface Specifications*.

Figure 7-1 illustrates the top-level signals and modules of the Custom PHY.

Figure 7-1. Custom PHY IP Core



For more detailed information about the Custom datapath and clocking, refer to the “Custom Configurations with the Standard PCS” section in the *Transceiver Custom Configurations in Stratix V Devices* chapter of the *Stratix V Device Handbook*.

Device Family Support

IP cores provide either final or preliminary support for target Altera device families. These terms have the following definitions:

- *Final support*—Verified with final timing models for this device.
- *Preliminary support*—Verified with preliminary timing models for this device.

Table 7-1 shows the level of support offered by the Custom PHY IP core for Altera device families.

Table 7-1. Device Family Support

Device Family	Support
Stratix V devices—Hard PCS and PMA	Preliminary
Arria V devices—Hard PCS and PMA	Preliminary
Cyclone V devices—Hard PCS and PMA	Preliminary
Other device families	No support

Performance and Resource Utilization

Because the PCS and PMA are both implemented in hard logic, the Custom PHY IP core requires less than 1% of FPGA resources. Table 7-1 lists the resource utilization for the Custom PHY when the 1.25GbE preset is specified.

Table 7-2. Custom PHY IP Core Performance and Resource Utilization—Stratix V GT Device

Channels	Combinational ALUTs	Logic Registers (Bits)
1	142	154
4	244	364

Parameter Settings

To configure the Custom PHY IP core in the parameter editor, click **Installed Plug-Ins > Interfaces > Transceiver PHY > Custom PHY v11.1**. You can use the tabs on the parameter editor to select the options required for the protocol. Presets are available for the 1.25 Gbps Ethernet (1.25GbE) protocol.

The following sections describe all of the options on seven tabs of the parameter editor and then list parameters that are set for 1.25GbE.

General Options

The **General Options** tab allows you to set the basic parameters of your transceiver PHY. Table 7–3 lists the settings available on the **General Options** tab.

Table 7–3. General Options (Part 1 of 3)

Name	Value	Description
Device family	Arria V Stratix V	Specifies the device family. Arria V and Stratix V are available.
Parameter validation rules	Custom GbE	Allows you to specify the transceiver protocol. Select Custom if you are not implementing GbE .
Mode of operation	Duplex TX RX	You can select to transmit data, receive data, or both.
Number of lanes	1–32	The total number of lanes in each direction.
Enable lane bonding	On/Off	When enabled, a single clock drives multiple lanes, reducing clock skew. In Stratix V devices, up to 6 lanes can be bonded if you use an ATX PLL; 5 lanes can be bonded if you select the CMU PLL.
Bonding mode	×N fb_compensation	Select ×N to use the same clock source for up to 6 channels in a single transceiver bank, resulting in reduced clock skew. You must use contiguous channels when you select ×N bonding. In addition, you must place logical channel 0 in either physical channel 1 or 4. Physical channels 1 and 4 are indirect drivers of the ×N clock network. Select fb_compensation (feedback compensation) to use the same clock source for multiple channels across different transceiver banks to reduce clock skew. For more information about bonding, refer to “Bonded Channel Configurations Using the PLL Feedback Compensation Path” in <i>Transceiver Clocking in Stratix V Devices</i> in volume 3 of the <i>Stratix V Device Handbook</i> .
FPGA fabric transceiver interface width	8,10,16,20, 32,40	Specifies the total serialization factor, from an input or output pin to the MAC-layer logic.

Table 7-3. General Options (Part 2 of 3)

Name	Value	Description																														
PCS-PMA interface width	8, 10, 16, 20	The PCS-PMA interface width depends on the FPGA fabric transceiver interface width and whether 8B/10B is enabled. The following combinations are available: <table> <tr> <th>FPGA/XCVR</th><th>8B/10B</th><th>PCS-PMA Interface Width</th></tr> <tr> <td>8</td><td>No</td><td>8</td></tr> <tr> <td>8</td><td>Yes</td><td>10</td></tr> <tr> <td>10</td><td>No</td><td>10</td></tr> <tr> <td>16</td><td>No</td><td>8 or 16</td></tr> <tr> <td>16</td><td>Yes</td><td>10 or 20</td></tr> <tr> <td>20</td><td>No</td><td>10 or 20</td></tr> <tr> <td>32</td><td>No</td><td>16</td></tr> <tr> <td>32</td><td>Yes</td><td>20</td></tr> <tr> <td>40</td><td>No</td><td>20</td></tr> </table>	FPGA/XCVR	8B/10B	PCS-PMA Interface Width	8	No	8	8	Yes	10	10	No	10	16	No	8 or 16	16	Yes	10 or 20	20	No	10 or 20	32	No	16	32	Yes	20	40	No	20
FPGA/XCVR	8B/10B	PCS-PMA Interface Width																														
8	No	8																														
8	Yes	10																														
10	No	10																														
16	No	8 or 16																														
16	Yes	10 or 20																														
20	No	10 or 20																														
32	No	16																														
32	Yes	20																														
40	No	20																														
PLL type	CMU ATX	You can select either the CMU or ATX PLL. The CMU PLL has a larger frequency range than the ATX PLL. The ATX PLL is designed to improve jitter performance and achieves lower channel-to-channel skew; however, it supports a narrower range of data rates and reference clock frequencies. Another advantage of the ATX PLL is that it does not use a transceiver channel, while the CMU PLL does. Because the CMU PLL is more versatile, it is specified as the default setting. An informational message displays in the message pane telling you whether the chosen settings for Data rate and Input clock frequency are legal for the CMU PLL, or for both the CMU and ATX PLLs.																														
Data rate	622–11000 Mbps	Specifies the data rate.																														
Base data rate	1 × Lane rate 2 × Lane rate 4 × Lane rate	The base data rate is the frequency of the clock input to the PLL. Select a base data rate that minimizes the number of PLLs required to generate all the clocks required for data transmission. By selecting an appropriate base data rate, you can change data rates by changing the divider used by the clock generation block. For higher frequency data rates 2 × and 4 × base data rates are not available.																														
Input clock frequency	Variable	Specifies the frequency of the PLL input reference clock. The frequency required is the Base data rate /2. You can use any Input clock frequency that allows the PLLs to generate this frequency.																														
Additional Options																																
Enable TX Bitflip	On/Off	When enabled, the TX bitflip word aligner is operational.																														
Create rx_coreclk_in port	On/Off	This is an optional clock to drive the coreclk of the RX PCS																														
Create tx_coreclk_in port	On/Off	This is an optional clock to drive the coreclk of the TX PCS																														
Create rx_recovered_clk port	On/Off	When enabled, the RX recovered clock is an output.																														
Create optional ports	On/Off	When you turn this option on, the following signals are added to the top level of your transceiver for each lane: ■ tx_forceidle ■ rx_is_lockedtoref ■ rx_is_lockedtodata ■ rx_signaldetect																														

Table 7-3. General Options (Part 3 of 3)

Name	Value	Description
Avalon data interfaces	On/Off	When you turn this option On , the order of symbols is changed. This option is typically required if you are planning to import your Custom PHY IP core into a Qsys system.
Enable embedded reset control	On/Off	When On, the automatic reset controller resets your design at power up. When Off you can design you own reset logic using tx_analogreset, rx_analogreset, tx_digitalreset, rx_digitalreset, and pll_powerdown which are top-level ports of the Custom Transceiver PHY. You must turn this option Off to implement your own reset controller. By default, the CDR circuitry is in automatic lock mode whether you use the embedded reset controller or design your own reset logic. You can switch the CDR to manual mode by writing the pma_rx_setlocktodata or pma_rx_set_locktoref registers to 1. If either the pma_rx_set_locktodata and pma_rx_set_locktoref is set, the CDR automatic lock mode is disabled as Table 7-4 illustrates. For more information about the reset control and status registers, refer to Table 7-21 on page 7-21. For more information about reset in Stratix V devices, refer to Transceiver Reset Control in Stratix V Devices in volume 3 of the <i>Stratix V Device Handbook</i>.

[Table 7-4](#) shows the resulting reset controller mode for various combinations of the pma_rx_setlocktodata and pma_rx_set_locktoref registers. As [Table 7-4](#) indicates, setting either pma_rx_setlocktodata or pma_rx_set_locktoref disables the automatic reset controller.

Table 7-4. Reset Mode

rx_set_locktoref	rx_set_locktodata	CDR Lock Mode
1	0	Manual RX CDR locked to reference
X	1	Manual RX CDR locked to data
0	0	Automatic RX CDR

Word Alignment

The word aligner restores word boundaries of received data based on a predefined alignment pattern. This pattern can be 7, 8, 10, 16, 20, or 32 bits long. The word alignment module searches for a programmed pattern to identify the correct boundary for the incoming stream. Table 7-5 lists the settings available on the **Word Aligner** tab.

Table 7-5. Word Aligner Options

Name	Value	Description
Word alignment mode	Manual	You can select 1 of the following 3 modes: ■ Manual—In this mode you enable the word alignment function by asserting <code>rx_enapatternalign</code> using the Avalon-MM interface. When the PCS exits reset, the word aligner automatically performs an initial alignment to the specified word alignment pattern. After the initial alignment, you must assert <code>rx_enapatternalign</code> to initiate another pattern alignment. <code>rx_enapatternalign</code> is edge sensitive. ■ Bit slipping—You can use bit slip mode to shift the word boundary using the Avalon-MM interface. For every rising edge of the <code>rx_bitslip</code> signal, the word boundary is shifted by 1 bit. Each bit slip removes the earliest received bit from the received data. ■ Automatic synchronization state machine—In this mode, word alignment is controlled by a programmable state machine. This mode can only be used with 8B/10B encoding. The data width at the word aligner can be 10 or 20 bits. You can specify the following parameters: ■ Number of consecutive valid words before sync state is reached: Specifies the number of consecutive valid words needed to reduce the built up error count by 1. Valid values are 1–256.
	Bit slipping	
	Automatic synchronization state machine	
Enable run length violation checking	On/Off	If you turn this option on, you can specify the run length which is the maximum legal number of contiguous 0s or 1s.
Run length	40–640	Specifies the threshold for a run-length violation.

Table 7–6 provides more information about the word alignment function.

Table 7–6. Word Aligner Options

Configuration	PMA-PCS Interface Width (bits)	Word Alignment Mode	Word Alignment Pattern Length (bits)	Word Alignment Behavior
Custom single deserializer width	8	Manual alignment	16	User-controlled signal starts alignment process. Alignment occurs once unless signal is re-asserted.
	10	Manual alignment	7, 10	User-controlled signal starts alignment process. Alignment occurs once unless signal is re-asserted.
		Automatic synchronized state machine		Data must be 8B/10B encoded and aligns to selected word aligner pattern.
Custom double deserializer width	16	Manual alignment	8, 16, 32	User-controlled signal starts alignment process. Alignment occurs once unless signal is re-asserted.
	20	Manual alignment	8, 16, 32	User-controlled signal starts alignment process. Alignment occurs once unless signal is re-asserted.
		Automatic Synchronized State Machine	7 and 10 bits	Automatically selected word aligner pattern length and pattern.

Rate Match FIFO

The rate match FIFO compensates for small clock frequency differences between the upstream transmitter and the local receiver clocks by inserting or removing skip (SKP) symbols or ordered-sets from the inter-packet gap (IPG) or idle streams. It deletes SKP symbols or ordered-sets when the upstream transmitter reference clock frequency is greater than the local receiver reference clock frequency. It inserts SKP symbols or ordered-sets when the local receiver reference clock frequency is greater than the upstream transmitter reference clock frequency.

If you enable the rate match FIFO, the parameter editor provides options to enter the rate match insertion and deletion patterns. The lower 10 bits are the control pattern, and the upper 10 bits are the skip pattern. Table 7–7 lists the settings available on the Rate Match tab.

Table 7–7. Rate Match FIFO Options (Part 1 of 2)

Name	Value	Description
Enable rate match FIFO	On/Off	Turn this option on, to enable the rate match functionality. Turning this option on adds the <code>rx_rmifodatainserted</code> , and <code>rx_rmifodatadeleted</code> status signals to your PHY.
Rate match insertion/deletion +ve disparity pattern	1101000011 1010000011	Enter a 10-bit skip pattern (bits 10–19) and a 10-bit control pattern (bits 0–9). The skip pattern must have neutral disparity.

Table 7–7. Rate Match FIFO Options (Part 2 of 2)

Name	Value	Description
Rate match insertion/deletion -ve disparity pattern	0010111100 0101111100	Enter a 10-bit skip pattern (bits 10–19) and a 10-bit control pattern (bits 0–9). The skip pattern must have neutral disparity.
Create optional rate match FIFO status ports	On/Off	When enabled, creates the <code>rx_rmifodddatainserted</code> and <code>rx_rmifodatadeleted</code> signals from the rate match FIFO become output ports.

8B/10B Encoder and Decoder

The 8B/10B encoder generates 10-bit code groups (control or data word) with proper disparity from the 8-bit data and 1-bit control identifier. The 8B/10B decoder receives 10-bit data from the rate matcher and decodes it into an 8-bit data and 1-bit control identifier. Table 7–8 lists the settings available on the **8B/10B** tab.

Table 7–8. 8B/10B Options

Name	Value	Description
Enable 8B/10B decoder/encoder	On/Off	Enable this option if your application requires 8B/10B encoding and decoding. This option on adds the <code>tx_dataak<n></code> , <code>rx_dataak<n></code> , and <code>rx_runningdisp<n></code> signals to your transceiver.
Enable manual disparity control	On/Off	When enabled, you can use the <code>tx_forcedisp</code> signal to control the disparity of the 8B/10B encoder. Turning this option on adds the <code>tx_forcedisp</code> and <code>tx_dispval</code> signals to your transceiver.
Create optional 8B/10B status port	On/Off	Enable this option to include the 8B/10B <code>rx_errdetect</code> and <code>rx_disperr</code> error signals at the top level of the Custom PHY IP core.

Byte Ordering

The byte ordering block is available when the PCS width is doubled at the byte deserializer. Byte ordering identifies the first byte of a packet by determining whether the programmed start-of-packet (SOP) pattern is present; it inserts enough pad characters in the data stream to force the SOP to the lowest order byte lane. Table 7–9 describes the byte order options.



You cannot enable Rate Match FIFO when your application requires byte ordering. Because the rate match function inserts and deletes idle characters, it may shift the SOP to a different byte lane.

Table 7–9. Byte Order Options

Name	Value	Description
Enable byte ordering block	On/Off	Turn this option on if your application uses serialization to create a datapath that is larger than 1 symbol. This option is only available if you use the byte deserializer.
Enable byte ordering block manual control	On/Off	Turn this option on to choose manual control of byte ordering. This option creates the <code>rx_enabyteord</code> signal. A byte ordering operation occurs whenever <code>rx_enabyteord</code> is asserted. To perform multiple byte ordering operations, deassert and reassert <code>rx_enabyteord</code> .

Table 7-9. Byte Order Options

Name	Value	Description												
Byte ordering pattern	11111011	Specifies the pattern that identifies the SOP. For 16-bit byte ordering pattern you must include a 2-bit pad so that the pattern entered is in the following format: 00 <pattern> 00 <pattern>. For example, if the required pattern is 10111100, enter the following pattern: 00101111000010111100												
Byte ordering pad pattern	00000000	Specifies the pad pattern that is inserted to align the SOP. Enter the following size pad patterns: <table> <tr> <th>Data Width</th><th>8B/10B Encoded?</th><th>Pad Pattern</th></tr> <tr> <td>8, 16, 32</td><td>No</td><td>8 bits</td></tr> <tr> <td>10,20,40</td><td>No</td><td>10 bits</td></tr> <tr> <td>8, 16, 32</td><td>Yes</td><td>9 bits</td></tr> </table>	Data Width	8B/10B Encoded?	Pad Pattern	8, 16, 32	No	8 bits	10,20,40	No	10 bits	8, 16, 32	Yes	9 bits
Data Width	8B/10B Encoded?	Pad Pattern												
8, 16, 32	No	8 bits												
10,20,40	No	10 bits												
8, 16, 32	Yes	9 bits												

PLL Reconfiguration

Table 7-10 lists the **PLL Reconfigurations** options. For more information about transceiver reconfiguration registers, refer to “**PLL Reconfiguration**” on page 10-18.

Table 7-10. PLL Reconfigurations

Name	Value	Description
Allow PLL Reconfiguration	On/Off	You must enable this option if you plan to reconfigure the PLLs in your design. This option is also required to simulate PLL reconfiguration.
Number of TX PLLs	1-4	Specifies the number of TX PLLs required for this instance of the Custom PHY. More than 1 PLL may be required if your design reconfigures channels to run at multiple frequencies.
Number of input clocks	1-5	Specifies the number of input reference clocks. More than one reference clock may be required if your design reconfigures channels to run at multiple frequencies.
Main TX PLL logical index	0-3	Specifies the index for the TX PLL that should be instantiated at startup. Logical index 0 corresponds to TX PLL0, and so on.
Main TX PLL input clock source	0-3	Specifies the index for the TX PLL input clock that should be instantiated at startup. Logical index 0 corresponds to input clock 0 and so on.
TX PLL (0-3) (Refer to Table 7-3 on page 7-3 for a detailed explanation of these parameters.)		
PLL Type	CMU ATX	Specifies the PLL type.
Base data rate	1 × Lane rate 2 × Lane rate 4 × Lane rate	Specifies Base data rate .
Input clock frequency	Variable	Specifies the frequency of the PLL input reference clock. The frequency required is the Base data rate /2. You can use any Input clock frequency that allows the PLLs to generate this frequency.
Selected input clock source	0-4	Specifies the index of the input clock for this TX PLL. Logical index 0 corresponds to input clock 0 and so on.

Table 7-10. PLL Reconfigurations

Name	Value	Description
Channel Interface		
Enable channel interface	On/Off	Turn this option on to enable PLL and datapath dynamic reconfiguration. When you select this option, the width of tx_parallel_data and rx_parallel_data buses increases in the following way. ■ The tx_parallel_data bus is 44 bits per lane; however, only the low-order number of bits specified by the FPGA fabric transceiver interface width contain valid data for each lane. ■ The rx_parallel_data bus is 64 bits per lane; however, only the low-order number of bits specified by the FPGA fabric transceiver interface width contain valid data.

Analog Options

You specify the analog parameters for Stratix V devices using the Quartus II Assignment Editor, the Pin Planner, or through the Quartus II Settings File (.qsf). The default values for analog options fall into three categories:

- *Global*—These parameters have default values that are independent of other parameter settings.
- *Computed*—These parameters have an initial default value that is recomputed based on other parameter settings.
- *Proxy*—These parameters have default values that are place holders. The Quartus II software selects these initial default values based on your design; however, Altera recommends that you replace these defaults with values that match your electrical board specification.

Table 7-11 lists the analog parameters for Stratix V devices whose original values are place holders for the values that match your electrical board specification. In Table 7-11, the default value of an analog parameter is shown in **bold** type.

Table 7-11. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 2)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_IO_PIN_TERMINATION	Transceiver I/O Pin Termination	Specifies the intended on-chip termination value for the specified transceiver pin. Use External Resistor if you intend to use off-chip termination.	85_OHMS 100_OHMS 120_OHMS 150_OHMS EXTERNAL_RESISTOR	Pin
XCVR_REFCLK_PIN_TERMINATION	Transceiver Dedicated Refclk Pin Termination	Specifies the intended termination value for the specified refclk pin.	DC_COUPLING_INTERNAL_100_OHM DC_COUPLING_EXTERNAL_RESISTOR AC_COUPLING	Pin

Table 7-11. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 2)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_RX_BYPASS_EQ_STAGES_234	Receiver Equalizer Stage 2, 3, 4 Bypass	Bypass continuous time equalizer stages 2, 3, and 4 to save power. This setting eliminates significant AC gain on the equalizer and is appropriate for chip-to-chip short range communication on a PCB.	ALL_STAGES_ENABLED BYPASS_STAGES	Pin
XCVR_TX_SLEW_RATE_CTRL	Transmitter Slew Rate Control	Specifies the slew rate of the output signal. The valid values span from the slowest rate to fastest rate with 1 representing the slowest rate.	1-5	Pin
XCVR_VCCA_VOLTAGE	VCCA_GXB Voltage	Configure the VCCA_GXB voltage for a GXB I/O pin by specifying the intended VCCA_GXB voltage for a GXB I/O pin. If you do not make this assignment the compiler automatically sets the correct VCCA_GXB voltage depending on the configured data rate, as follows: ■ Data rate <= 6.5 Gbps: 2_5V ■ Data rate > 6.5 Gbps: 3_0V or 3_3V for Stratix V ES silicon	2_5V 3_0V	Pin
XCVR_VCCR_VCCT_VOLTAGE	VCCR_GXB VCCT_GXB Voltage	Configure the VCCR_GXB and VCCT_GXB voltage for an GXB I/O pin by specifying the intended supply voltages for a GXB I/O pin. If your design uses decision feedback equalization (DFE) or adaptive equalization (AEQ), you must set this parameter to 1.0V. Otherwise, if you do not make this assignment the compiler automatically sets the correct VCCR_GXB and VCCT_GXB voltage depending on the configured data rate as follows: ■ Data rate <= 6.5 Gbps: 0_85V ■ Data rate > 6.5 Gbps: 1_0V	0_85V 1_0V	Pin

Table 7–12 lists the analog parameters with *global* or *computed* default values. You may want to optimize some of these settings. In Table 7–12, the default value is shown in **bold** type. For computed analog parameters, the default value listed is for the initial setting, not the recomputed setting.

Table 7–12. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 3)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
Analog Parameters with Global Default Value				
CDR_BANDWIDTH_PRESET	CDR Bandwidth Preset	Specifies the CDR bandwidth preset setting.	Auto Low Medium High	PLL instance
PLL_BANDWIDTH_PRESET	PLL Bandwidth Preset	Specifies the PLL bandwidth preset setting	Auto Low Medium High	PLL instance
XCVR_RX_DC_GAIN	Receiver Buffer DC Gain Control	Controls the amount of a stage receive-buffer DC gain.	0 –4	Pin
XCVR_RX_LINEAR_EQUALIZER_CONTROL	Receiver Linear Equalizer Control	Static control for the continuous time equalizer in the receiver buffer. The equalizer has 16 distinct settings from 0–15 corresponding to the increasing AC gain.	1 –16	Pin
Analog Parameter with Computed Default Value				
XCVR_RX_COMMON_MODE_VOLTAGE	Receiver Buffer Common Mode Voltage	Receiver buffer common-mode voltage.	VTT_OP80V VTT_OP75V VTT_OP70V VTT_OP65V VTT_OP60V VTT_OP55V VTT_OP50V VTT_OP35V VTT_PUP _WEAK VTT_PDN WEAK TRISTATE1 VTT_PDN_ STRONG VTT_PUP_ STRONG TRISTATE2 TRISTATE3 TRISTATE4	Pin
XCVR_RX_ENABLE_LINEAR_EQUALIZER_PCIEMODE	Receiver Linear Equalizer Control (PCI Express)	If enabled equalizer gain control is driven by the PCS block for PCI Express. If disabled equalizer gain control is determined by the XCVR_RX_LINEAR_EQUALIZER_SETTING assignment.	TRUE FALSE	Pin

Table 7-12. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 3)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_RX_EQ_BW_SEL	Receiver Equalizer Gain Bandwidth Select	Sets the gain peaking frequency for the equalizer. For data-rates of less than 6.5Gbps set to HALF. For higher data-rates set to FULL.	FULL HALF	Pin
XCVR_RX_SD_ENABLE	Receiver Signal Detection Unit Enable/Disable	Enables or disables the receiver signal detection unit.	TRUE FALSE	Pin
XCVR_RX_SD_OFF	Receiver Cycle Count Before Signal Detect Block Declares Loss Of Signal	Number of parallel cycles to wait before the signal detect block declares loss of signal.	0–29	Pin
XCVR_RX_SD_ON	Receiver Cycle Count Before Signal Detect Block Declares Presence Of Signal	Number of parallel cycles to wait before the signal detect block declares presence of signal.	0–16	Pin
XCVR_RX_SD_THRESHOLD	Receiver Signal Detection Voltage Threshold	Specifies signal detection voltage threshold level.	0 –7	Pin
XCVR_TX_COMMON_MODE_VOLTAGE	Transmitter Common Mode Driver Voltage	Transmitter common-mode driver voltage	VOLT_OP80V VOLT_OP75V VOLT_OP70V VOLT_OP65V VOLT_OP60V VOLT_OP55V VOLT_OP50V VOLT_OP35V PULL_UP PULL_DOWN TRISTATED1 GROUNDED PULL_UP_TO_VCCELA TRISTATED2 TRISTATED3 TRISTATED4	Pin
XCVR_TX_PRE_EMP_1ST_POST_TAP	Transmitter Preemphasis First Post-Tap	Specifies the first post-tap setting value.	0 –31	Pin
XCVR_TX_PRE_EMP_2ND_POST_TAP	Transmitter Preemphasis Second Post-Tap	Specifies the second post-tap setting value.	0–15	Pin
XCVR_TX_PRE_EMP_INV_2ND_TAP	Transmitter Preemphasis Second Tap Invert	Inverts the transmitter pre-emphasis 2nd post tap.	TRUE FALSE	Pin
XCVR_TX_PRE_EMP_INV_PRE_TAP	Transmitter Preemphasis Pre Tap Invert	Inverts the transmitter pre-emphasis pre-tap.	TRUE FALSE	Pin
XCVR_TX_PRE_EMP_PRE_TAP	Transmitter Preemphasis Pre-Tap	Specifies the pre-tap pre-emphasis setting.	0 –15	Pin
XCVR_TX_RX_DET_ENABLE	Transmitter's Receiver Detect Block Enable	Enables or disables the receiver detector circuit at the transmitter.	TRUE FALSE	Pin
XCVR_TX_RX_DET_MODE	Transmitter's Receiver Detect Block Mode	Sets the mode for receiver detect block	0–15	Pin
XCVR_TX_RX_DET_OUTPUT_SEL	Transmitter's Receiver Detect Block QPI/PCI Express Control	Determines QPI or PCI Express mode for the Receiver Detect block.	RX_DET_QPI_OUT RX_DET_PCIE_OUT	Pin

Table 7–12. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 3)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_TX_VOD	Transmitter Differential Output Voltage	Differential output voltage setting. The values are monotonically increasing with the driver main tap current strength.	0–63 50	Pin
XCVR_TX_VOD_PRE_EMP_CTRL_SRC	Transmitter V _{OD} /Preemphasis Control Source	When set to DYNAMIC_CTL, the PCS block controls the V _{OD} and preemphasis coefficients for PCI Express. When this assignment is set to RAM_CTL the V _{OD} and preemphasis are controlled by other assignments, such as XCVR_TX_PRE_EMP_1ST_POST_TAP.	DYNAMIC_CTL RAM_CTL	Pin

❓ For more information about the Pin Planner, refer to [About the Pin Planner](#) in Quartus II Help. For more information about the Assignment Editor, refer to [About the Assignment Editor](#) in Quartus II Help.

 For more information about Quartus II Settings, refer to [Quartus II Settings File Manual](#).

Presets for Ethernet

If you apply the presets for 1.25GbE, parameters with specific required values for those protocols are set for you. Once applied, the preset is in boldface and remains as such unless you change some of the preset parameters. Selecting a preset does not prevent you from changing any parameter to meet the requirements of your design. [Table 7–13](#) lists the parameters that are set for the 1.25GbE protocol.

Table 7–13. Presets for the 1.25GbE Protocol

Parameter Name	1.25GbE
General Options Tab	
Transceiver Protocol	GbE
Enable bonding	Off
FPGA fabric transceiver interface width	8
Data rate	1250 Mbps
Input clock frequency	125 MHz
Enable TX Bitslip	Off
Create rx_coreclkkin port	Off
Create tx_coreclkkin port	Off
Create optional ports	Off
8B/10B Options	
Enable 8B/10B decoder/encoder	On

Table 7-13. Presets for the 1.25GbE Protocol

Parameter Name	1.25GbE
Enable manual disparity control	Off
Create optional 8B/10B status port	Off
Word Aligner Options	
Word alignment mode	Automatic synchronization state machine
Number of consecutive valid words before sync state is reached	1
Number of bad data words before loss of sync state	1
Number of valid patterns before sync state is reached	10
Create optional word aligner status ports	Off
Word aligner pattern length	10
Word alignment pattern	1011111100
Enable run length violation checking	Off
Run length	40
Rate Match Options	
Enable rate match FIFO	On
Rate match insertion/deletion +ve disparity pattern	1010001001010111100
Rate match insertion/deletion -ve disparity pattern	10101011011010000011
Byte Order Options	
Enable byte ordering block	Off
Enable byte ordering block manual control	Off
Byte ordering pattern	1111111011
Byte ordering pad pattern	N/A
Datapath	
Deserializer block width	Auto

Interfaces

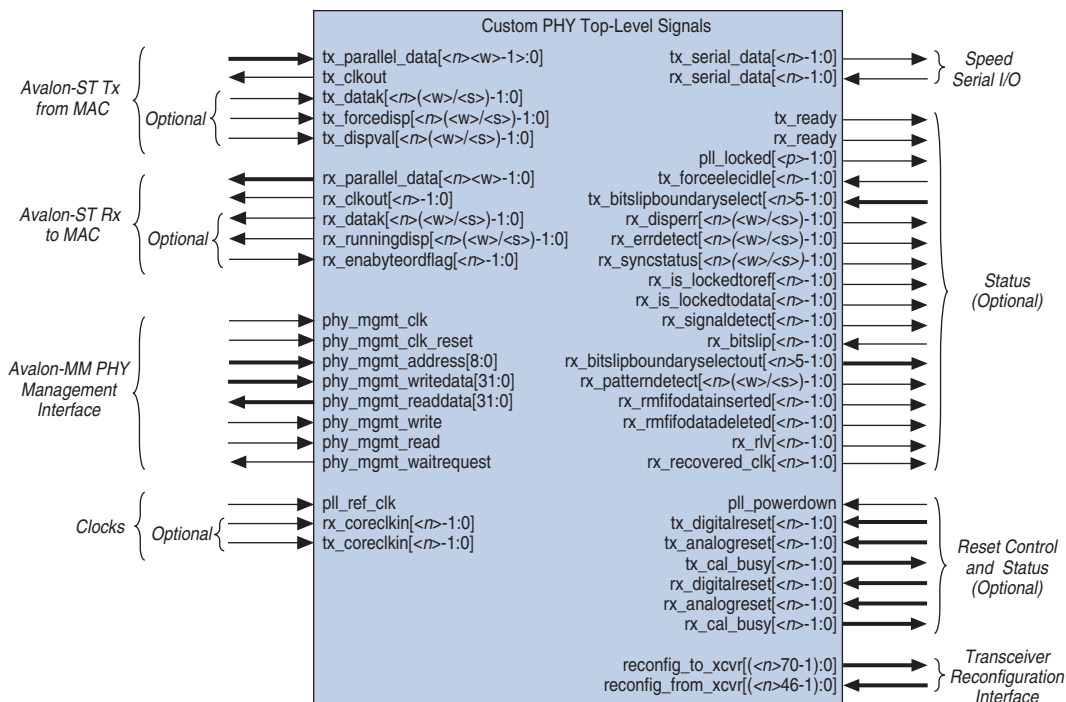
This section describes interfaces of the Custom Transceiver PHY. It includes the following topics:

- [Ports](#)
- [Register Interface](#)
- [Dynamic Reconfiguration](#)

Ports

Figure 7-2 illustrates the top-level signals of the Custom PHY IP core. The variables in Figure 7-2 represent the following parameters:

- $\langle n \rangle$ —The number of lanes
- $\langle w \rangle$ —The width of the FPGA fabric to transceiver interface per lane
- $\langle s \rangle$ —The symbol size
- $\langle p \rangle$ —The number of PLLs

Figure 7-2. Custom PHY Top-Level Signals (1)**Note to Figure 7-2:**

(1) $\langle n \rangle$ is the number of lanes. $\langle w \rangle$ is the PCS to FPGA fabric interface width per lane. $\langle s \rangle$ is the symbol size in bits. $\langle p \rangle$ is the number of PLLs.



By default **block diagram** shown in the parameter editor labels the external pins with the *interface type* and places the *interface name* inside the box. The interface type and name are used in the `_hw.tcl` file that describes the component. If you turn on **Show signals**, the **block diagram** displays all top-level signal names.



For more information about `_hw.tcl` files refer to the [Component Interface Tcl Reference](#) chapter in volume 1 of the *Quartus II Handbook*.

The following sections describe the signals in each interface.

Avalon-ST TX Input Data from the MAC

Table 7-14 describes the signals in the Avalon-ST input interface. These signals are driven from the MAC to the PCS. This is an Avalon sink interface.



For more information about the Avalon-ST protocol, including timing diagrams, refer to the *Avalon Interface Specifications*.

Table 7-14. Avalon-ST TX Interface

Signal Name	Direction	Description
tx_parallel_data[(<n><w>)-1:0]	Sink	This is TX parallel data driven from the MAC. The ready latency on this interface is 0, so that the PHY must be able to accept data as soon as it comes out of reset.
tx_clkout	Output	This is the clock for TX parallel data, control, and status signals.
tx_dataak[(<n>(<w>/<s>)-1:0]	Sink	Data and control indicator for the received data. When 0, indicates that tx_data is data, when 1, indicates that tx_data is control.
tx_forcedisp[(<n>(<w>/<s>)-1:0]	Sink	When asserted, this control signal enables disparity to be forced on the TX channel. This signal is created if you turn On the Enable manual disparity control option on the 8B/10B tab.
tx_dispval[(<n>(<w>/<s>)-1:0]	Sink	This control signal specifies the disparity of the data. This port is created if you turn On the Enable disparity control option on the 8B/10B tab.

Avalon-ST RX Output Data to the MAC

Table 7-15 describes the signals in the Avalon-ST output interface. These signals are driven from the PCS to the MAC. This is an Avalon source interface.

Table 7-15. Avalon-ST RX Interface

Signal Name	Direction	Description
rx_parallel_data[(<n><w>)-1:0]	Source	This is RX parallel data driven from the Custom PHY IP core. The ready latency on this interface is 0, so that the MAC must be able to accept data as soon as the PHY comes out of reset. Data driven from this interface is always valid.
rx_clkout[(<n>)-1:0]	Output	This is the clock for the RX parallel data source interface.
rx_dataak[(<n>(<w>/<s>)-1:0]	Source	Data and control indicator for the source data. When 0, indicates that rx_parallel_data is data, when 1, indicates that rx_parallel_data is control.
rx_runningdisp[(<n>(<w>/<s>)-1:0]	Source	This status signal indicates the disparity of the incoming data.
rx_enabyteordflag[(<n>)-1:0]	Input	This signal is created if you turn On the Enable byte ordering block control option on the Byte Order tab. A byte ordering operation occurs whenever rx_enabyteord is asserted. To perform multiple byte ordering operations, deassert and reassert rx_enabyteordflag.

Clock Interface

Table 7-16 describes optional and required clocks for the Custom PHY. The input reference clock, `pll_ref_clk`, drives a PLL inside the PHY-layer block, and a PLL output clock, `rx_clkout` (described in Table 7-15 on page 7-17) is used for all data, command, and status inputs and outputs.

Table 7-16. Clock Signals

Signal Name	Direction	Description
<code>pll_ref_clk</code>	Input	Reference clock for the PHY PLLs. Frequency range is 50–700 MHz.
<code>rx_coreclkln[<n>-1:0]</code>	Input	This is an optional clock to drive the coreclk of the RX PCS.
<code>tx_coreclkln[<n>-1:0]</code>	Input	This is an optional clock to drive the coreclk of the TX PCS

Transceiver Serial Data Interface

Table 7-17 describes the differential serial data interface and the status signals for the RX interface.

Table 7-17. Serial Interface and Status Signals ⁽¹⁾

Signal Name	Direction	Signal Name
<code>rx_serial_data[<n>-1:0]</code>	Input	Receiver differential serial input data.
<code>tx_serial_data[<n>-1:0]</code>	Output	Transmitter differential serial output data.

Note to Table 7-17:

(1) <n> is the number of lanes. <w> is the PCS to FPGA fabric interface width. <s> is the symbol size in bits. <p> is the number of PLLs.

Status Signals (Optional)

Table 7-18 describes the optional status signals for the RX interface.

Table 7-18. Serial Interface and Status Signals (Part 1 of 2) ⁽¹⁾

Signal Name	Direction	Signal Name
<code>tx_ready</code>	Output	When asserted, indicates that the TX interface has exited the reset state and is ready to transmit.
<code>rx_ready</code>	Output	When asserted, indicates that the RX interface has exited the reset state and is ready to receive.
<code>pll_locked[<p>-1:0]</code>	Output	When asserted, indicates that the PLL is locked to the input reference clock.
<code>tx_forceelecidle[<n>-1:0]</code>	Input	When asserted, enables a circuit to detect a downstream receiver. It is used for the PCI Express protocol. This signal must be driven low when not in use because it causes the TX PMA to enter electrical idle mode and tristate the TX serial data signals.
<code>tx_bitslipboundaryselect[<n>5-1:0]</code>	Input	This signal is used for bit slip word alignment mode. It selects the number of bits that the TX block must slip to achieve a deterministic latency.
<code>rx_disper[<n>(<w>/<s>)-1:0]</code>	Output	When asserted, indicates that the received 10-bit code or data group has a disparity error.
<code>rx_errdetect[<n>(<w>/<s>)-1:0]</code>	Output	When asserted, indicates that a received 10-bit code group has an 8B/10B code violation or disparity error.

Table 7-18. Serial Interface and Status Signals (Part 2 of 2)⁽¹⁾

Signal Name	Direction	Signal Name
rx_syncstatus[<n>(<w>/<s>)-1:0]	Output	Indicates presence or absence of synchronization on the RX interface. Asserted when word aligner identifies the word alignment pattern or synchronization code groups in the received data stream. This signal is optional.
rx_is_lockedtoref[<n>-1:0]	Output	Asserted when the receiver CDR is locked to the input reference clock. This signal is asynchronous. This signal is optional.
rx_is_lockedtodata[<n>-1:0]	Output	When asserted, the receiver CDR is in to lock-to-data mode. When deasserted, the receiver CDR lock mode depends on the rx_locktorefclk signal level. This signal is optional.
rx_signaldetect[<n>-1:0]	Output	Signal threshold detect indicator required for the PCI Express protocol. When asserted, it indicates that the signal present at the receiver input buffer is above the programmed signal detection threshold value.
rx_bitslip[<n>-1:0]	Input	Used for manual control of bit slipping. The word aligner slips a bit of the current word for every rising edge of this signal.
rx_bitslipboundaryselectout[<n>5-1:0]	Output	This signal is used for bit slip word alignment mode. It reports the number of bits that the RX block slipped to achieve a deterministic latency.
rx_patterndetect[<n>(<w>/<s>)-1:0]	Output	When asserted, indicates that the programmed word alignment pattern has been detected in the current word boundary.
rx_rmifodatainserted[<n>-1:0]	Output	When asserted, indicates that the RX rate match block inserted an R column.
rx_rmifodatadeleted[<n>-1:0]	Output	When asserted, indicates that the RX rate match block deleted an R column.
rx_rlv[<n>-1:0]	Output	When asserted, indicates a run length violation. Asserted if the number of consecutive 1s or 0s exceeds the number specified in the parameter editor.
rx_recovered_clk[<n>-1:0]	Output	This is the RX clock which is recovered from the received data stream.

Note to Table 7-17:

(1) <n> is the number of lanes. <w> is the PCS to FPGA fabric interface width per lane. <s> is the symbol size in bits. <p> is the number of PLLs.

Reset Control and Status (Optional)

Table 7-19 describes the signals in the optional reset control and status interface. These signals are available if you do not enable the embedded reset controller. For more information including timing diagrams, refer to *Transceiver Reset Control in Stratix V Devices* in volume 3 of the *Stratix V Device Handbook*.

Table 7-19. Avalon-ST RX Interface (Part 1 of 2)

Signal Name	Direction	Description
pll_powerdown	Input	When asserted, resets the TX PLL.
tx_digitalreset[<n>-1:0]	Input	When asserted, reset all blocks in the TX PCS.
tx_analogreset[<n>-1:0]	Input	When asserted, resets all blocks in the TX PMA.
tx_cal_busy[<n>-1:0] >	Output	When asserted, indicates that the TX channel is being calibrated. You must hold the channel in reset until calibration completes.

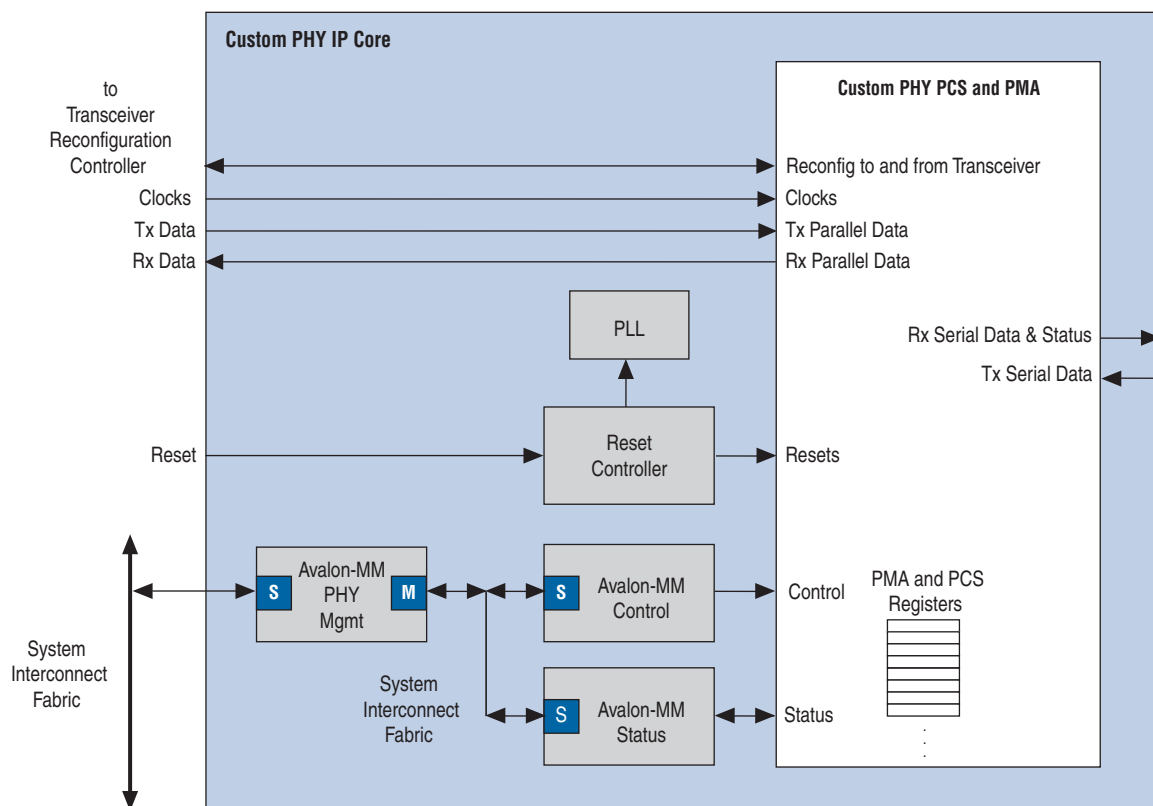
Table 7-19. Avalon-ST RX Interface (Part 2 of 2)

Signal Name	Direction	Description
rx_digitalreset[<n>-1:0]	Input	When asserted, resets the RX PCS.
rx_analogreset[<n>-1:0]	Input	When asserted, resets the RX CDR.
rx_cal_busy[<n>-1:0]	Output	When asserted, indicates that the RX channel is being calibrated. You must hold the channel in reset until calibration completes.

Register Interface

The Avalon-MM PHY management interface provides access to the Custom PHY PCS and PMA registers, resets, error handling, and serial loopback controls. You can use an embedded controller acting as an Avalon-MM master to send read and write commands to this Avalon-MM slave interface.

Figure 7-3 provides a high-level view of this hardware.

Figure 7-3. Custom PHY IP Core (1)

Note to Figure 7-3:

(1) Blocks in gray are soft logic. Blocks in white are hard logic.

Table 7–20 describes the signals in the PHY Management interface.

Table 7–20. Avalon-MM PHY Management Interface

Signal Name	Direction	Description
phy_mgmt_clk	Input	Avalon-MM clock input. There is no frequency restriction for the phy_mgmt_clk; however, if you plan to use the same clock for the PHY management interface and transceiver reconfiguration, you must restrict the frequency range of phy_mgmt_clk to 100–125 MHz to meet the specification for the transceiver reconfiguration clock.
phy_mgmt_clk_reset	Input	Global reset signal. This signal is active high and level sensitive.
phy_mgmt_address[8:0]	Input	9-bit Avalon-MM address.
phy_mgmt_writedata[31:0]	Input	Input data.
phy_mgmt_readdata[31:0]	Output	Output data.
phy_mgmt_write	Input	Write signal.
phy_mgmt_read	Input	Read signal.
phy_mgmt_waitrequest	Output	When asserted, indicates that the Avalon-MM slave interface is unable to respond to a read or write request. When asserted, control signals to the Avalon-MM slave interface must remain constant.

Register Descriptions

Table 7–21 specifies the registers that you can access over the PHY management interface using word addresses and a 32-bit embedded processor. A single address space provides access to all registers.



Writing to reserved or undefined register addresses may have undefined side effects.

Table 7–21. Custom PHY IP Core Registers (Part 1 of 3)

Word Addr	Bits	R/W	Register Name	Description
PMA Common Control and Status Registers				
0x022	[31:0]	R	pma_tx_pll_is_locked	Bit[P] indicates that the TX/CMU PLL (P) is locked to the input reference clock. There is typically one pma_tx_pll_is_locked bit per system.
Reset Control Registers—Automatic Reset Controller				
0x041	[31:0]	RW	reset_ch_bitmask	Reset controller channel bitmask for digital resets. The default value is all 1s. Channel <n> can be reset when bit<n> = 1.

Table 7–21. Custom PHY IP Core Registers (Part 2 of 3)

Word Addr	Bits	R/W	Register Name	Description
0x042	[1:0]	W	reset_control (write)	Writing a 1 to bit 0 initiates a TX digital reset using the reset controller module. The reset affects channels enabled in the reset_ch_bitmask. Writing a 1 to bit 1 initiates a RX digital reset of channels enabled in the reset_ch_bitmask.
		R	reset_status(read)	Reading bit 0 returns the status of the reset controller TX ready bit. Reading bit 1 returns the status of the reset controller RX ready bit.
Reset Controls –Manual Mode				
0x044	[31:0]	RW	reset_fine_control	You can use the reset_fine_control register to create your own reset sequence. If you disable Enable embedded reset controller on the General Options tab of the parameter editor, you can design your own reset sequence using the tx_analogreset, rx_analogreset, tx_digitalreset, rx_digitalreset, and pll_powerdown which are top-level ports of the Custom Transceiver PHY. By default, the CDR circuitry is in automatic lock mode whether you use the embedded reset controller or design your own reset logic. You can switch the CDR to manual mode by writing the pma_rx_setlocktodata or pma_rx_set_locktoref registers to 1.
	[31:4,0]	RW	Reserved	It is safe to write 0s to reserved bits.
	[3]	RW	reset_rx_digital	Writing a 1 causes the internal RX digital reset signal to be asserted, resetting the RX digital channels enabled in reset_ch_bitmask. You must write a 0 to clear the reset condition.
	[2]	RW	reset_rx_analog	Writing a 1 causes the internal RX analog reset signal to be asserted, resetting the RX analog logic of all channels enabled in reset_ch_bitmask. You must write a 0 to clear the reset condition.
	[1]	RW	reset_tx_digital	Writing a 1 causes the internal TX digital reset signal to be asserted, resetting all channels enabled in reset_ch_bitmask. You must write a 0 to clear the reset condition.
PMA Control and Status Registers				
0x061	[31:0]	RW	phy_serial_loopback	Writing a 1 to channel <n> puts channel <n> in serial loopback mode. For information about pre- or post-CDRserial loopback modes, refer to “ Loopback Modes ” on page 10–39 .
0x063	[31:0]	R	pma_rx_signaldetect	When channel <n> =1, indicates that receive circuit for channel <n> senses the specified voltage exists at the RX input buffer.
0x064	[31:0]	RW	pma_rx_set_locktodata	When set, programs the RX CDR PLL to lock to the incoming data. Bit <n> corresponds to channel <n>.
0x065	[31:0]	RW	pma_rx_set_locktoref	When set, programs the RX CDR PLL to lock to the reference clock. Bit <n> corresponds to channel <n>.

Table 7–21. Custom PHY IP Core Registers (Part 3 of 3)

Word Addr	Bits	R/W	Register Name	Description
0x066	[31:0]	R	pma_rx_is_lockedtoata	When 1, indicates that the RX CDR PLL is locked to the RX data, and that the RX CDR has changed from LTR to LTD mode. Bit <n> corresponds to channel <n>.
0x067	[31:0]	R	pma_rx_is_lockedtoref	When 1, indicates that the RX CDR PLL is locked to the reference clock. Bit <n> corresponds to channel <n>.
Custom PCS				
0x080	[31:0]	RW	Lane or group number	Specifies lane or group number for indirect addressing, which is used for all PCS control and status registers. For variants that stripe data across multiple lanes, this is the logical group number. For non-bonded applications, this is the logical lane number.
0x081	[5:1]	R	rx_bitslipboundaryselect out	This is an output from the bit slip word aligner which shows the number of bits slipped. From block: Word aligner.
	[0]	R	rx_phase_comp_fifo_error	When set, indicates an RX phase compensation FIFO error. From block: RX phase Compensation FIFO
0x082	[0]	RW	tx_phase_comp_fifo_error	When set, indicates an TX phase compensation FIFO error. From block: TX phase Compensation FIFO
0x083	[5:1]	RW	tx_bitslipboundary_select	Sets the number of bits that the TX bit slipper needs to slip. To block: Word aligner.
	[0]	RW	tx_invpolarity	When set, the TX interface inverts the polarity of the TX data. To block: 8B/10B encoder.
0x084	0	RW	rx_invpolarity	When set, the RX channels inverts the polarity of the received data. To block: 8B/10B decoder.
0x085	[3]	RW	rx_bitslip	Every time this register transitions from 0 to 1, the RX data slips a single bit. To block: Word aligner.
	[2]	RW	rx_bytereversal_enable	When set, enables byte reversal on the RX interface. To block: Byte deserializer.
	[1]	RW	rx_bitreversal_enable	When set, enables bit reversal on the RX interface. To block: Word aligner.
	[0]	RW	rx_enapatternalign	When set in manual word alignment mode, the word alignment logic begins operation when this pattern is set. To block: Word aligner.

Dynamic Reconfiguration

As silicon progresses towards smaller process nodes, circuit performance is affected more by variations due to process, voltage, and temperature (PVT). These process variations result in analog voltages that can be offset from required ranges. The calibration performed by the dynamic reconfiguration interface compensates for variations due to PVT.

For Stratix V devices, each channel and each TX PLL have separate dynamic reconfiguration interfaces. The parameter editor provides informational messages on the connectivity of these interfaces. [Example 7-1](#) shows the messages for a single duplex channel parameterized for the 1.25GbE protocol.

Although you must initially create a separate reconfiguration interface for each channel and TX PLL in your design, when the Quartus II software compiles your design, it reduces the number of reconfiguration interfaces by merging reconfiguration interfaces. The synthesized design typically includes a reconfiguration interface for three channels. Allowing the Quartus II software to merge reconfiguration interfaces gives the Fitter more flexibility in placing transceiver channels. For more information about transceiver reconfiguration refer to [Chapter 10, Transceiver Reconfiguration Controller](#).

Example 7-1. Informational Messages for the Transceiver Reconfiguration Interface

PHY IP will require 2 reconfiguration interfaces for connection to the external reconfiguration controller.
 Reconfiguration interface offset 0 is connected to the transceiver channel.
 Reconfiguration interface offset 1 is connected to the transmit PLL.

[Table 7-22](#) describes the signals in the reconfiguration interface. This interface uses the Avalon-MM PHY Management interface clock.

Table 7-22. Reconfiguration Interface

Signal Name	Direction	Description
reconfig_to_xcvr [$(\langle n \rangle - 1) : 0$]	Sink	Reconfiguration signals from the Transceiver Reconfiguration Controller. $\langle n \rangle$ grows linearly with the number of reconfiguration interfaces.
reconfig_from_xcvr [$(\langle n \rangle - 1) : 0$]	Source	Reconfiguration signals to the Transceiver Reconfiguration Controller. $\langle n \rangle$ grows linearly with the number of reconfiguration interfaces.

Simulation Files and Example Testbench

Refer to [“Running a Simulation Testbench” on page 1-4](#) for a description of the directories and files that the Quartus II software creates automatically when you generate your Custom PHY IP core.



Refer to the [Altera wiki](#) for an example testbench that you can use as a starting point in creating your own verification environment.

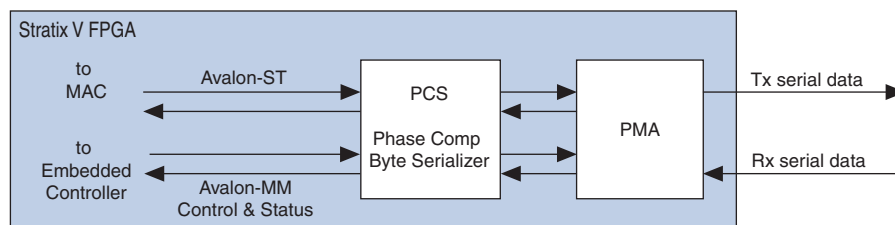
The Altera Low Latency PHY IP core receives and transmits differential serial data, recovering the RX clock from the RX input stream. The PMA connects to a simplified PCS whose single function doubles the width of the TX and RX datapaths. Depending on the configuration you choose, the Low Latency PHY IP core instantiates one of the following channels:

- GX channels using the Standard PCS with a frequency range of 1–8.5 Gbps
- GX channels using the 10G PCS with a frequency range of 1–12.5 Gbps
- GT channels in PMA-Direct mode with a maximum frequency of 20–28 Gbps. The maximum frequency for Stratix V ES silicon is 26 Gbps.

An Avalon-ST interface is used for TX and RX data for the MAC interface. An Avalon-MM interface provides access to control and status information.

Figure 8–1 illustrates the top-level modules of the Low Latency PHY IP core.

Figure 8–1. Low-Latency PHY IP Core—Stratix V Devices



Because the Low latency PHY IP core bypasses much of the PCS, it minimizes the PCS latency.



For more detailed information about the Low Latency datapath and clocking, refer to the “Stratix V GX Device Configurations” section in the *Transceiver Configurations in Stratix V Devices* chapter of the *Stratix V Device Handbook*.

Device Family Support

IP cores provide either final or preliminary support for target Altera device families. These terms have the following definitions:

- *Final support*—Verified with final timing models for this device.
- *Preliminary support*—Verified with preliminary timing models for this device.

Table 8–1 shows the level of support offered by the PMA IP core for Altera device families.

Table 8–1. Device Family Support

Device Family	Support
Stratix V devices	Preliminary
Other device families	No support

Performance and Resource Utilization

Table 8–2 shows the typical expected device resource utilization for different configurations using the current version of the Quartus II software targeting a Stratix V GX (5SGSMD612H35C2) device.

Table 8–2. Low Latency PHY Performance and Resource Utilization—Stratix V GX Device

Implementation	Number of Lanes	Serialization Factor	Worst-Case Frequency	Combinational ALUTs	Dedicated Registers	Memory Bits
11 Gbps	1	32 or 40	599.16	112	95	0
11 Gbps	4	32 or 40	584.8	141	117	0
11 Gbps	10	32 or 40	579.71	192	171	0
6 Gbps (10 Gbps datapath)	1	32 or 40	608.27	111	93	0
6 Gbps (10 Gbps datapath)	4	32 or 40	454.96	141	117	0
6 Gbps (10 Gbps datapath)	10	32 or 40	562.75	192	171	0
6 Gbps (8 Gbps datapath)	1	32 or 40	607.16	113	93	0
6 Gbps (8 Gbps datapath)	4	32 or 40	639.8	142	117	0
6 Gbps (8 Gbps datapath)	10	32 or 40	621.89	193	171	0
3 Gbps (8 Gbps datapath)	1	8, 10, 16, or 20	673.4	114	93	0
3 Gbps (8 Gbps datapath)	4	8, 10, 16, or 20	594.88	142	117	0
3 Gbps (8 Gbps datapath)	10	8, 10, 16, or 20	667.67	193	171	0

Parameter Settings

To configure the Low Latency PHY IP core in the MegaWizard Plug-In Manager, click **Installed Plug-Ins > Interfaces > Transceiver PHY > Low Latency PHY v11.1**. For more information about using the MegaWizard Plug-In Manager refer to [Chapter 2, Getting Started](#).

General Options

[Table 8–3](#) lists the settings available on **General Options** tab.

Table 8–3. General Options

Name	Value	Description
Device family	Stratix V	This IP core is only available for Stratix V.
Datapath type	Standard 10G GT	The Low Latency PHY IP core is part of a Standard , 10G , or GT datapath. In most cases the FPGA fabric transceiver interface width determines the bandwidth of the datapath; however, when the FPGA fabric transceiver interface width is 32 or 40 bits, you have the option of using either the Standard datapath which is the default mode, or changing to the 10G datapath by selecting this option. Refer to Table 8–4 for a comprehensive list of datapath support.
Mode of operation	Duplex RX TX	Specifies the mode of operation as Duplex , RX , or TX mode.
Number of lanes	1–32 1–4	Specifies the total number of lanes in each direction. Stratix V devices include up to 32 GX channels (Standard or 10G) and up to 4 GT channels. You must instantiate each GT channel in a separate Low Latency PHY IP core instance. You cannot specify both GX and GT channels within the same instance.
Enable lane bonding	On/Off	When enabled, the PMA uses the same clock source for up to 6 channels in a transceiver bank, reducing clock skew. This option is only available for the Standard datapath. Turn this option Off if you are using multiple TX PLLs in a single Low Latency PHY IP core instance.
Bonding mode	×N fb_compensation	Select ×N to use the same clock source for up to 6 channels in a single transceiver bank, resulting in reduced clock skew. You must use contiguous channels when you select ×N bonding. In addition, you must place logical channel 0 in either physical channel 1 or 4. Physical channels 1 and 4 are indirect drivers of the ×N clock network. Select fb_compensation (feedback compensation) to use the same clock source for multiple channels across different transceiver banks to reduce clock skew. For more information about bonding, refer to “Bonded Channel Configurations Using the PLL Feedback Compensation Path” in Transceiver Clocking in Stratix V Devices in volume 3 of the <i>Stratix V Device Handbook</i> .

Table 8-3. General Options

Name	Value	Description
FPGA fabric transceiver interface width	8, 10, 16, 20, 32, 40, 50, 64, 66, 128	This option indicates the parallel data fabric transceiver interface width. GT datapath supports a single width of 128 bits. Refer to Table 8-4 for the supported interface widths of the Standard and 10G datapaths.
PCS-PMA interface width	8, 10, 16, 20, 32, 30, 64	The PCS-PMA interface width depends on the FPGA fabric transceiver interface width and the Datapath type. Refer to Table 8-4 for the supported interface widths of the Standard and 10G datapaths.
PLL type	CMU ATX	The CMU PLL is available for the Standard and 10G datapaths. The ATX PLL is available for the Standard , 10G , and GT datapaths. The CMU PLL has a larger frequency range than the ATX PLL. The ATX PLL is designed to improve jitter performance and achieves lower channel-to-channel skew; however, it supports a narrower range of data rates and reference clock frequencies. Another advantage of the ATX PLL is that it does not use a transceiver channel, while the CMU PLL does. An informational message displays in the message panel if the PLL type that you select is not available at the frequency specified.
Data rate	Standard: 1–8.5 Gbps 10G: 1–12.5 Gbps GT: 20–28 Gbps	Specifies the data rate in Mbps.
Base data rate	1 × Lane rate 2 × Lane rate 4 × Lane rate	Select a base data rate that minimizes the number of PLLs required to generate all the clocks required for data transmission. By selecting an appropriate base data rate , you can change data rates by changing the divider used by the clock generation block. For higher frequency data rates 2 × and 4 × base data rates are not available.
Input clock frequency	Variable	Specifies the frequency of the PLL input reference clock. The frequency required is the Base data rate /2. You can use any Input clock frequency that allows the PLLs to generate this frequency.

Table 8-4 lists **Standard** and **10G** datapath widths for the FPGA fabric-transceiver interface, the PCS-PMA interface, and the resulting frequencies for the tx_clkout and rx_clkout parallel clocks. In almost all cases, the parallel clock frequency is described by the following equation:

$$frequency_{parallel\ clock} = data\ rate / FPGA\ fabric-transceiver\ interface\ width$$



The FPGA fabric-transceiver interface width is always 128 bits for the **GT** datapath.

Table 8-4. Datapath Width Support (Part 1 of 2)

FPGA Fabric-Transceiver Interface Width	PCS-PMA Interface Width		tx_clkout and rx_clkout frequency
	Standard Datapath	10G Datapath	
8	8	—	data rate/8
10	10	—	data rate/10

Table 8–4. Datapath Width Support (Part 2 of 2)

FPGA Fabric-Transceiver Interface Width	PCS-PMA Interface Width		tx_clkout and rx_clkout frequency
	Standard Datapath	10G Datapath	
16	8 or 16	—	data rate/16
20	10 or 20	—	data rate/20
32	16	32	data rate/32
40	20	40	data rate/40
50	—	40	data rate/50 ⁽¹⁾
64	—	32	data rate/32 ⁽²⁾
64	—	64	data rate/64
66	—	40	data rate/66

Note to Table 8–4:

- (1) For this datapath configuration, the tx_clkout frequency generated by the the Low Latency PHY is the **data rate** /40. You must generate a /50 frequency clock from the /40 clock and feed this clock back into the tx_coreclkin. The rx_clkout frequency generated by the Low Latency PHY is /40 of the data rate. You must generate a /50 frequency from the recovered clock and feed this back into the rx_coreclkin.
- (2) For this datapath configuration, the tx_clkout frequency generated by the the Low Latency PHY is the **data rate** /32. You must generate a /64 frequency clock from the /32 clock and feed this clock back into the tx_coreclkin. The rx_clkout frequency generated by the Low Latency PHY is the data rate/32. You must generate a /64 frequency from the recovered clock and feed this back into the rx_coreclkin.

Additional Options

The parameters on the **Additional Options** tab control clocking and datapath options. Both bonded (×N) and non-bonded modes are available. In bonded modes, a single PLL can drive all channels. In non-bonded modes, each channel may have its own PLL.

Table 8–5 describes the options available on the **Additional Options** tab.

Table 8–5. Additional Options (Part 1 of 2)

Name	Value	Description
Enable tx_coreclkin ⁽¹⁾	On/Off	When you turn this option on, tx_coreclkin connects to the write clock of the TX phase compensation FIFO and you can clock the parallel TX data generated in the FPGA fabric using this port. This port allows you to clock the write side of the TX phase compensation FIFO with a user-provided clock, either the FPGA fabric clock, the FPGA fabric-TX interface clock, or the input reference clock. You must turn this option On when the FPGA fabric transceiver interface width:PCS-PMA Interface width is 50:40 or when you specify the 10G datapath with a fabric transceiver interface width:PCS-PMA Interface width of 64:32. For the GT datapath, if you are using different reference clock pins for the TX and RX channels, you must instantiate two separate Low Latency PHY IP core instances for TX and RX channels. The reference clock pins for each channel must reside in the same transceiver bank.

Table 8–5. Additional Options (Part 2 of 2)

Name	Value	Description
Enable rx_coreclkin ⁽¹⁾	On/Off	When you turn this option on, rx_coreclkin connects to the read clock of the RX phase compensation FIFO and you can clock the parallel RX output data using rx_coreclk. This port allows you to clock the read side of the RX phase compensation FIFO with a user-provided clock, either the FPGA fabric clock, the FPGA fabric RX interface clock, or the input reference clock. rx_coreclkin is not available for the GT datapath. You must turn this option On when the FPGA fabric transceiver interface width:PCS-PMA Interface width is 50:40 or when you specify the 10G datapath with a fabric transceiver interface width:PCS-PMA Interface width of 64:32.
Enable TX bit slip	On/Off	When you turn this option on, the variant includes the tx_bitslip signal. For every rising edge of the tx_bitslip signal, the word boundary is shifted by 1 bit. Each bit slip removes the earliest received bit from the received data. This option is only available for the Standard and 10G datapath.
Enable embedded reset control	On/Off	This option is turned on by default. When On, the embedded reset controller initiates the reset sequence when it receives a positive edge on the phy_mgmt_clk_reset input signal. For more information about the embedded reset controller, refer to the “Embedded Reset Controller” section in the <i>Transceiver Reset Control in Stratix V Devices</i> in volume 3 of the <i>Stratix V Device Handbook</i>. Disable this option to implement your own reset sequence using the tx_analogreset, rx_analogreset, tx_digitalreset, rx_digitalreset, and pll_powerdown which are available as top-level ports of the Low Latency Transceiver PHY. When you design your own reset controller, the tx_ready and rx_ready are not top-level signals of the core. For more information about designing a reset controller, refer to the “User-Controller Reset Controller” section in the <i>Transceiver Reset Control in Stratix V Devices</i> in volume 3 of the <i>Stratix V Device Handbook</i>.
Avalon data interfaces	On/Off	When you turn this option On, the order of symbols is changed. This option is typically required if you are planning to import your Low Latency Transceiver PHY IP core into a Qsys system.
Avalon data symbol size	All bits 8 10 16 20	Specifies the number of bits that are in the smallest unit of data. One or more symbols make up a single unit of data transferred in a cycle.

Note to Table 8–5:

(1) For more information refer to the “FPGA Fabric-Transceiver Interface Clocking” section in the *Stratix V Transceiver Clocking* chapter.

PLL Reconfiguration Options

Table 8–5 describes the options available on the **PLL Reconfiguration** tab. For more information about transceiver reconfiguration registers, refer to “**PLL Reconfiguration**” on page 10–18.

 The PLL reconfiguration options are not available for the GT datapath.

Table 8–6. PLL Reconfigurations

Name	Value	Description
Allow PLL Reconfiguration	On/Off	You must enable this option if you plan to reconfigure the PLLs in your design. This option is also required to simulate PLL reconfiguration.
Number of TX PLLs	1–4	Specifies the number of TX PLLs required for this instance of the Low Latency Transceiver PHY. More than 1 PLL may be required if your design reconfigures channels to run at multiple frequencies.
Number of input clocks	1–5	Specifies the number of input reference clocks. More than one reference clock may be required if your design reconfigures channels to run at multiple frequencies.
Main TX PLL logical index	0–3	Specifies the index for the TX PLL that should be instantiated at startup. Logical index 0 corresponds to TX PLL0, and so on.
CDR PLL input clock source	0–3	Specifies the index for the TX PLL input clock that should be instantiated at startup. Logical index 0 corresponds to input clock 0 and so on.
TX PLL (0–3) (Refer to Table 8–3 on page 8–3 for a detailed explanation of these parameters.)		
PLL Type	CMU ATX	Specifies the PLL type.
Base data rate	1 × Lane rate 2 × Lane rate 4 × Lane rate	Specifies Base data rate .
Input clock frequency	Variable	Specifies the frequency of the PLL input reference clock. The frequency required is the Base data rate /2. You can use any Input clock frequency that allows the PLLs to generate this frequency.
Selected input clock source	0–4	Specifies the index of the input clock for this TX PLL. Logical index 0 corresponds to input clock 0 and so on.
Channel Interface		
Enable Channel Interface	On/Off	Turn this option on to enable PLL and datapath dynamic reconfiguration. When you select this option, the width of tx_parallel_data and rx_parallel_data buses increases in the following way. ■ Standard datapath: ■ The tx_parallel_data bus is 44 bits per lane; however, only the low-order number of bits specified by the FPGA fabric transceiver interface width contain valid data for each lane. ■ The rx_parallel_data bus is 64 bits per lane; however, only the low-order number of bits specified by the FPGA fabric transceiver interface width contain valid data. ■ 10G datapath: ■ The both the tx_parallel_data and rx_parallel_data buses are 64 bits per lane; however, only the low-order number of bits specified by the FPGA fabric transceiver interface width contain valid data.

Analog Options

You specify the analog parameters for Stratix V devices using the Quartus II Assignment Editor, the Pin Planner, or through the Quartus II Settings File (.qsf). The default values for analog options fall into three categories:

- *Global*—These parameters have default values that are independent of other parameter settings.
- *Computed*—These parameters have an initial default value that is recomputed based on other parameter settings.
- *Proxy*—These parameters have default values that are place holders. The Quartus II software selects these initial default values based on your design; however, Altera recommends that you replace these defaults with values that match your electrical board specification.

Table 8-7 lists the analog parameters for Stratix V devices whose original values are place holders for the values that match your electrical board specification. In Table 8-7, the default value of an analog parameter is shown in **bold** type.

Table 8-7. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 2)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_IO_PIN_TERMINATION	Transceiver I/O Pin Termination	Specifies the intended on-chip termination value for the specified transceiver pin. Use External Resistor if you intend to use off-chip termination.	85_OHMS 100_OHMS 120_OHMS 150_OHMS EXTERNAL_RESISTOR	Pin
XCVR_REFCLK_PIN_TERMINATION	Transceiver Dedicated Refclk Pin Termination	Specifies the intended termination value for the specified refclk pin.	DC_COUPLING_INTERNAL_100_OHM DC_COUPLING_EXTERNAL_RESISTOR AC_COUPLING	Pin
XCVR_RX_BYPASS_EQ_STAGES_234	Receiver Equalizer Stage 2, 3, 4 Bypass	Bypass continuous time equalizer stages 2, 3, and 4 to save power. This setting eliminates significant AC gain on the equalizer and is appropriate for chip-to-chip short range communication on a PCB.	ALL_STAGES_ENABLED BYPASS_STAGES	Pin
XCVR_TX_SLEW_RATE_CTRL	Transmitter Slew Rate Control	Specifies the slew rate of the output signal. The valid values span from the slowest rate to fastest rate with 1 representing the slowest rate.	1-5	Pin

Table 8–7. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 2)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_VCCA_VOLTAGE	VCCA_GXB Voltage	Configure the VCCA_GXB voltage for a GXB I/O pin by specifying the intended VCCA_GXB voltage for a GXB I/O pin. If you do not make this assignment the compiler automatically sets the correct VCCA_GXB voltage depending on the configured data rate, as follows: ■ Data rate <= 6.5 Gbps: 2_5V ■ Data rate > 6.5 Gbps: 3_0V or 3_3V for Stratix V ES silicon	2_5V 3_0V	Pin
XCVR_VCCR_VCCT_VOLTAGE	VCCR_GXB VCCT_GXB Voltage	Configure the VCCR_GXB and VCCT_GXB voltage for an GXB I/O pin by specifying the intended supply voltages for a GXB I/O pin. If your design uses decision feedback equalization (DFE) or adaptive equalization (AEQ), you must set this parameter to 1.0V. Otherwise, if you do not make this assignment the compiler automatically sets the correct VCCR_GXB and VCCT_GXB voltage depending on the configured data rate as follows: ■ Data rate <= 6.5 Gbps: 0_85V ■ Data rate > 6.5 Gbps: 1_0V	0_85V 1_0V	Pin

Table 8–8 lists the analog parameters with *global* or *computed* default values. You may want to optimize some of these settings. In Table 8–8, the default value is shown in **bold** type. For computed analog parameters, the default value listed is for the initial setting, not the recomputed setting.

Table 8–8. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 3)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
Analog Parameters with Global Default Value				
CDR_BANDWIDTH_PRESET	CDR Bandwidth Preset	Specifies the CDR bandwidth preset setting.	Auto Low Medium High	PLL instance
PLL_BANDWIDTH_PRESET	PLL Bandwidth Preset	Specifies the PLL bandwidth preset setting	Auto Low Medium High	PLL instance
XCVR_RX_DC_GAIN	Receiver Buffer DC Gain Control	Controls the amount of a stage receive-buffer DC gain.	0–4	Pin

Table 8–8. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 3)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_RX_LINEAR_EQUALIZER_CONTROL	Receiver Linear Equalizer Control	Static control for the continuous time equalizer in the receiver buffer. The equalizer has 16 distinct settings from 0–15 corresponding to the increasing AC gain.	1–16	Pin
Analog Parameter with Computed Default Value				
XCVR_RX_COMMON_MODE_VOLTAGE	Receiver Buffer Common Mode Voltage	Receiver buffer common-mode voltage.	VTT_OP80V VTT_OP75V VTT_OP70V VTT_OP65V VTT_OP60V VTT_OP55V VTT_OP50V VTT_OP35V VTT_PUP_WEAK VTT_PDN_WEAK TRISTATE1 VTT_PDN_STRONG VTT_PUP_STRONG TRISTATE2 TRISTATE3 TRISTATE4	Pin
XCVR_RX_ENABLE_LINEAR_EQUALIZER_PCIEMODE	Receiver Linear Equalizer Control (PCI Express)	If enabled equalizer gain control is driven by the PCS block for PCI Express. If disabled equalizer gain control is determined by the XCVR_RX_LINEAR_EQUALIZER_SETTING assignment.	TRUE FALSE	Pin
XCVR_RX_EQ_BW_SEL	Receiver Equalizer Gain Bandwidth Select	Sets the gain peaking frequency for the equalizer. For data-rates of less than 6.5Gbps set to HALF. For higher data-rates set to FULL.	FULL HALF	Pin
XCVR_RX_SD_ENABLE	Receiver Signal Detection Unit Enable/Disable	Enables or disables the receiver signal detection unit.	TRUE FALSE	Pin
XCVR_RX_SD_OFF	Receiver Cycle Count Before Signal Detect Block Declares Loss Of Signal	Number of parallel cycles to wait before the signal detect block declares loss of signal.	0 –29	Pin
XCVR_RX_SD_ON	Receiver Cycle Count Before Signal Detect Block Declares Presence Of Signal	Number of parallel cycles to wait before the signal detect block declares presence of signal.	0 –16	Pin
XCVR_RX_SD_THRESHOLD	Receiver Signal Detection Voltage Threshold	Specifies signal detection voltage threshold level.	0 –7	Pin

Table 8–8. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 3)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_TX_COMMON_MODE_VOLTAGE	Transmitter Common Mode Driver Voltage	Transmitter common-mode driver voltage	VOLT_OP80V VOLT_OP75V VOLT_OP70V VOLT_OP65V VOLT_OP60V VOLT_OP55V VOLT_OP50V VOLT_OP35V PULL_UP PULL_DOWN TRISTATED1 GROUNDED PULL_UP_TO_VCCELA TRISTATED2 TRISTATED3 TRISTATED4	Pin
XCVR_TX_PRE_EMP_1ST_POST_TAP	Transmitter Preemphasis First Post-Tap	Specifies the first post-tap setting value.	0 –31	Pin
XCVR_TX_PRE_EMP_2ND_POST_TAP	Transmitter Preemphasis Second Post-Tap	Specifies the second post-tap setting value.	0 –15	Pin
XCVR_TX_PRE_EMP_INV_2ND_TAP	Transmitter Preemphasis Second Tap Invert	Inverts the transmitter pre-emphasis 2nd post tap.	TRUE FALSE	Pin
XCVR_TX_PRE_EMP_INV_PRE_TAP	Transmitter Preemphasis Pre Tap Invert	Inverts the transmitter pre-emphasis pre-tap.	TRUE FALSE	Pin
XCVR_TX_PRE_EMP_PRE_TAP	Transmitter Preemphasis Pre-Tap	Specifies the pre-tap pre-emphasis setting.	0 –15	Pin
XCVR_TX_RX_DET_ENABLE	Transmitter's Receiver Detect Block Enable	Enables or disables the receiver detector circuit at the transmitter.	TRUE FALSE	Pin
XCVR_TX_RX_DET_MODE	Transmitter's Receiver Detect Block Mode	Sets the mode for receiver detect block	0 –15	Pin
XCVR_TX_RX_DET_OUTPUT_SEL	Transmitter's Receiver Detect Block QPI/PCI Express Control	Determines QPI or PCI Express mode for the Receiver Detect block.	RX_DET_QPI_OUT RX_DET_PCIE_OUT	Pin
XCVR_TX_VOD	Transmitter Differential Output Voltage	Differential output voltage setting. The values are monotonically increasing with the driver main tap current strength.	0–63 50	Pin
XCVR_TX_VOD_PRE_EMP_CTRL_SRC	Transmitter V _{OD} /Preemphasis Control Source	When set to DYNAMIC_CTL, the PCS block controls the V _{OD} and preemphasis coefficients for PCI Express. When this assignment is set to RAM_CTL the V _{OD} and preemphasis are controlled by other assignments, such as XCVR_TX_PRE_EMP_1ST_POST_TAP.	DYNAMIC_CTL RAM_CTL	Pin

For more information about the Pin Planner, refer to [About the Pin Planner](#) in Quartus II Help. For more information about the Assignment Editor, refer to [About the Assignment Editor](#) in Quartus II Help.

For more information about Quartus II Settings, refer to [Quartus II Settings File Manual](#).

Interfaces

This section describes interfaces of the Low Latency Transceiver PHY. It includes the following topics:

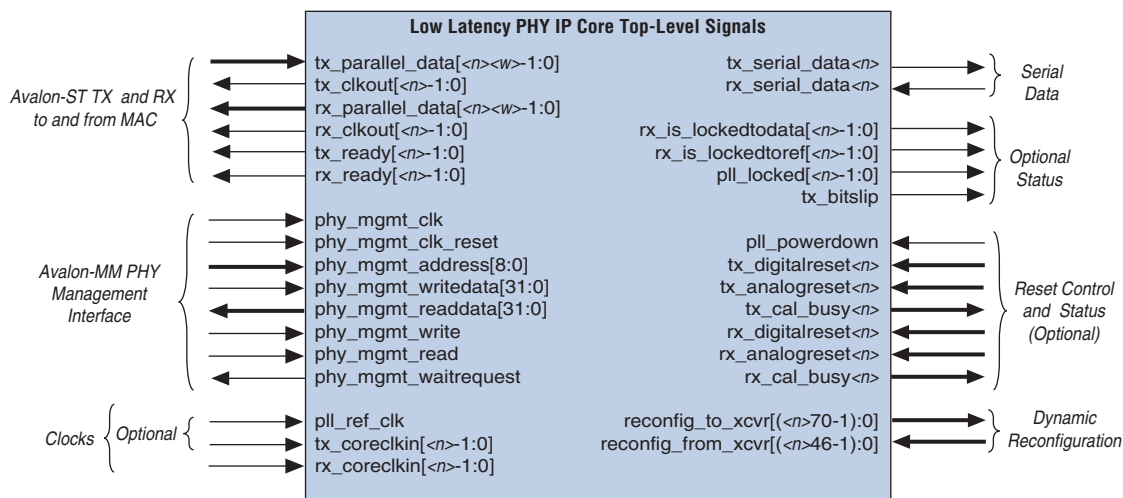
- [Ports](#)
- [Register Interface](#)
- [Dynamic Reconfiguration](#)

Ports

Figure 8-2 illustrates the top-level signals of the Custom PHY IP core. The variables in Figure 8-2 represent the following parameters:

- $\langle n \rangle$ —The number of lanes
- $\langle w \rangle$ —The width of the FPGA fabric to transceiver interface per lane

Figure 8-2. Top-Level Low Latency Signals



By default **block diagram** shown in the parameter editor labels the external pins with the *interface type* and places the *interface name* inside the box. The interface type and name are used in the `_hw.tcl` file that describes the component. If you turn on **Show signals**, the **block diagram** displays all top-level signal names.



For more information about `_hw.tcl` files refer to the *Component Interface Tcl Reference* chapter in volume 1 of the *Quartus II Handbook*.

The following sections describe each interface.

Avalon-ST TX and RX Data Interface to the FPGA Fabric

Table 8-9 describes the signals in the Avalon-ST interface. These signals are named from the point of view of the MAC so that the TX interface is an Avalon-ST sink interface and the RX interface is an Avalon-ST source.

Table 8-9. Avalon-ST interface

Signal Name	Direction	Description
<code>tx_parallel_data [<n><w>-1:0]</code>	Sink	This is TX parallel data driven from the MAC FPGA fabric. The ready latency on this interface is 0, so that the PCS in Low-Latency Bypass Mode or the MAC in PMA Direct mode must be able to accept data as soon as it comes out of reset.
<code>tx_clkout [<n>-1:0]</code>	Output	This is the clock for TX parallel data.
<code>tx_ready [<n>-1:0]</code>	Output	When asserted, indicates that the Low Latency IP core has exited the reset state is ready to receive data from the MAC. This signal is available if you select Enable embedded reset control on the Additional Options tab.
<code>rx_parallel_data [<n><w>-1:0]</code>	Source	This is RX parallel data driven by the Low Latency PHY IP core. Data driven from this interface is always valid.
<code>rx_clkout [<n>-1:0]</code>	Output	Low speed clock recovered from the serial data.
<code>rx_ready [<n>-1:0]</code>	Output	This is the ready signal for the RX interface. The ready latency on this interface is 0, so that the MAC must be able to accept data as soon as the PMA comes out of reset. This signal is available if you select Enable embedded reset control on the Additional Options tab.

Serial Data Interface

Table 8-10 describes the signals that comprise the serial data interface.

Table 8-10. Serial Data Interface

Signal Name	Direction	Description
<code>rx_serial_data [<n>-1:0]</code>	Sink	Differential high speed input serial data.
<code>tx_serial_data [<n>-1:0]</code>	Source	Differential high speed output serial data.

Optional Status Interface

Table 8–11 describes the signals that comprise the optional status interface.

Table 8–11. Optional Status Interface

Signal Name	Direction	Description
<code>rx_is_lockedtodata[<n>-1:0]</code>	Output	When asserted, indicates that the RX CDR is locked to incoming data. This signal is optional. If latency is not critical, you can read the value of this signal from the <code>Rx_is_lockedtodata</code> register.
<code>rx_is_lockedtoref[<n>-1:0]</code>	Output	When asserted, indicates that the RX CDR is locked to the input reference clock. This signal is optional. When the RX CDR is locked to data, you can ignore transitions on this signal. If latency is not critical, you can read the value of this signal from the <code>rx_is_lockedtoref</code> register.
<code>pll_locked[<n>-1:0]</code>	Output	When asserted, indicates that the TX PLL is locked to the input reference clock. This signal is asynchronous.
<code>tx_bitslip[<n>-1:0]</code>	Output	When set, the data sent to the PMA is slipped. The maximum number of bits that can be slipped is equal to the value selected in the serialization factor field - 1 or <code><d> - 1</code> .

Clock Interface

Table 8–12 describes reference clock for the Low Latency PHY. The input reference clock, `pll_ref_clk`, drives a PLL inside the PHY-layer block, and a PLL output clock, `rx_clkout` is used for all data, command, and status inputs and outputs.

Table 8–12. Clock Signals

Signal Name	Direction	Description
<code>tx_coreclk[<n>-1:0]</code>	Input	This is an optional clock to drive the write side of the TX FIFO.
<code>rx_coreclk[<n>-1:0]</code>	Input	This is an optional clock to drive the read side of the RX FIFO.
<code>pll_ref_clk</code>	Input	Reference clock for the PHY PLLs. The frequency range is 60–700 MHz.

Reset Control and Status (Optional)

Table 8–13 describes the signals in the optional reset control and status interface. These signals are available if you do not enable the embedded reset controller. For more information including timing diagrams, refer to *Transceiver Reset Control in Stratix V Devices* in volume 3 of the *Stratix V Device Handbook*.

Table 8–13. Avalon-ST RX Interface (Part 1 of 2)

Signal Name	Direction	Description
<code>pll_powerdown</code>	Input	When asserted, resets the TX PLL.
<code>tx_digitalreset[<n>-1:0]</code>	Input	When asserted, reset all blocks in the TX PCS.
<code>tx_analogreset[<n>-1:0]</code>	Input	When asserted, resets all blocks in the TX PMA.
<code>tx_cal_busy[<n>-1:0]</code>	Output	When asserted, indicates that the TX channel is being calibrated. You must hold the channel in reset until calibration completes.
<code>rx_digitalreset[<n>-1:0]</code>	Input	When asserted, resets the RX PCS.

Table 8-13. Avalon-ST RX Interface (Part 2 of 2)

Signal Name	Direction	Description
rx_analogreset[<n>-1:0]	Input	When asserted, resets the RX CDR.
rx_cal_busy[<n>-1:0]	Output	When asserted, indicates that the RX channel is being calibrated. You must hold the channel in reset until calibration completes.

Register Interface

The Avalon-MM PHY management interface provides access to the Low Latency PHY PCS and PMA registers that control the TX and RX channels, the PMA powerdown, PLL registers, and loopback modes. Figure 8-3 provides a high-level view of this hardware.

Figure 8-3. PMA

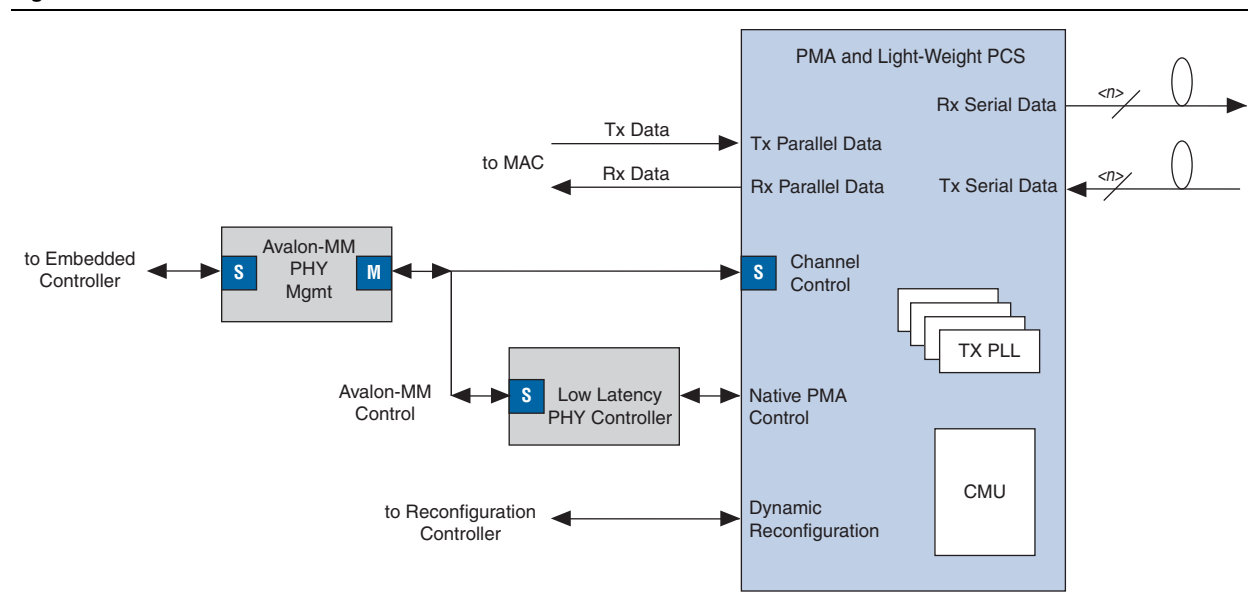


Table 8-14 describes the signals in the PHY Management interface.

Table 8-14. Avalon-MM PHY Management Interface (Part 1 of 2)

Signal Name	Direction	Description
phy_mgmt_clk	Input	Avalon-MM clock input. There is no frequency restriction for the phy_mgmt_clk; however, if you plan to use the same clock for the PHY management interface and transceiver reconfiguration, you must restrict the frequency range of phy_mgmt_clk to 100–125 MHz to meet the specification for the transceiver reconfiguration clock.
phy_mgmt_clk_reset	Input	Global reset signal. This signal is active high and level sensitive. This is an asynchronous signal.
phy_mgmtaddress[8:0]	Input	9-bit Avalon-MM address.
phy_mgmt_writedata[31:0]	Input	Input data.
phy_mgmt_readdata[31:0]	Output	Output data.

Table 8-14. Avalon-MM PHY Management Interface (Part 2 of 2)

Signal Name	Direction	Description
phy_mgmt_write	Input	Write signal.
phy_mgmt_read	Input	Read signal.



For more information about the Avalon-MM and Avalon-ST protocols, including timing diagrams, refer to the [Avalon Interface Specifications](#).

Register Descriptions

[Table 8-15](#) describes the registers that you can access over the PHY Management Interface using word addresses and a 32-bit embedded processor.



Writing to reserved or undefined register addresses may have undefined side effects.

Table 8-15. Low Latency PHY IP Core Registers (Part 1 of 2)

Word Addr	Bits	R/W	Register Name	Description
Reset Control Registers—Automatic Reset Controller				
0x041	[31:0]	RW	reset_ch_bitmask	Reset controller channel bitmask for digital resets. The default value is all 1s. Channel $\langle n \rangle$ can be reset when bit $\langle n \rangle = 1$.
0x042	[1:0]	W	reset_control (write)	Writing a 1 to bit 0 initiates a TX digital reset using the reset controller module. The reset affects channels enabled in the reset_ch_bitmask. Writing a 1 to bit 1 initiates a RX digital reset of channels enabled in the reset_ch_bitmask.
		R	reset_status(read)	Reading bit 0 returns the status of the reset controller TX ready bit. Reading bit 1 returns the status of the reset controller RX ready bit.
PMA Control and Status Registers				
0x061	[31:0]	RW	phy_serial_loopback	Writing a 1 to channel $\langle n \rangle$ puts channel $\langle n \rangle$ in serial loopback mode. For information about pre- or post-CDRserial loopback modes, refer to “ Loopback Modes ” on page 10–39.
0x063	[31:0]	R	pma_rx_signaldetect	When channel $\langle n \rangle = 1$, indicates that receive circuit for channel $\langle n \rangle$ senses the specified voltage exists at the RX input buffer.
0x064	[31:0]	RW	pma_rx_set_locktodata	When set, programs the RX CDR PLL to lock to the incoming data. Bit $\langle n \rangle$ corresponds to channel $\langle n \rangle$.
0x065	[31:0]	RW	pma_rx_set_locktoref	When set, programs the RX CDR PLL to lock to the reference clock. Bit $\langle n \rangle$ corresponds to channel $\langle n \rangle$.

Table 8-15. Low Latency PHY IP Core Registers (Part 2 of 2)

Word Addr	Bits	R/W	Register Name	Description
0x066	[31:0]	R	pma_rx_is_lockedtodata	When asserted, indicates that the RX CDR PLL is locked to the RX data, and that the RX CDR has changed from LTR to LTD mode. Bit $\langle n \rangle$ corresponds to channel $\langle n \rangle$.
0x067	[31:0]	R	pma_rx_is_lockedtoref	When asserted, indicates that the RX CDR PLL is locked to the reference clock. Bit $\langle n \rangle$ corresponds to channel $\langle n \rangle$.

Dynamic Reconfiguration

As silicon progresses towards smaller process nodes, circuit performance is affected more by variations due to process, voltage, and temperature (PVT). These process variations result in analog voltages that can be offset from required ranges. The calibration performed by the dynamic reconfiguration interface compensates for variations due to PVT.

For Stratix V devices, each channel and each TX PLL have separate dynamic reconfiguration interfaces.

Although you must initially create a separate reconfiguration interface for each channel and TX PLL in your design, when the Quartus II software compiles your design, it reduces the number of reconfiguration interfaces by merging reconfiguration interfaces. The synthesized design typically includes a reconfiguration interface for three channels. Allowing the Quartus II software to merge reconfiguration interfaces gives the Fitter more flexibility in placing transceiver channels. For more information about transceiver reconfiguration refer to [Chapter 10, Transceiver Reconfiguration Controller](#).

[Table 8-16](#) describes the signals in the reconfiguration interface. This interface uses a clock provided by the reconfiguration controller.

Table 8-16. Reconfiguration Interface

Signal Name	Direction	Description
reconfig_to_xcvr [($\langle n \rangle > 70$) - 1:0]	Sink	Reconfiguration signals from the Transceiver Reconfiguration Controller. $\langle n \rangle$ grows linearly with the number of reconfiguration interfaces.
reconfig_from_xcvr [($\langle n \rangle > 46$) - 1:0]	Source	Reconfiguration signals to the Transceiver Reconfiguration Controller. $\langle n \rangle$ grows linearly with the number of reconfiguration interfaces.

Simulation Files and Example Testbench

Refer to [“Running a Simulation Testbench” on page 1-4](#) for a description of the directories and files that the Quartus II software creates automatically when you generate your Low Latency PHY IP core.



Refer to the [Altera wiki](#) for an example testbench that you can use as a starting point in creating your own verification environment.

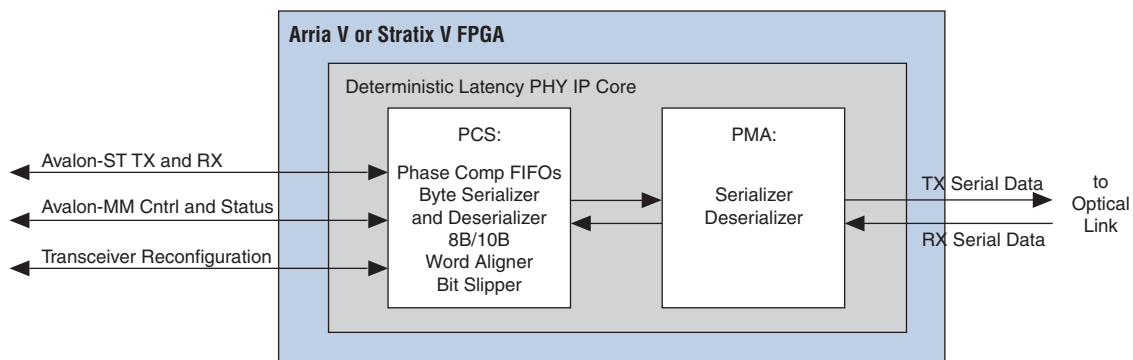
The Altera Deterministic Latency PHY IP Core targets protocols that require a datapath with deterministic latency. Deterministic latency enables accurate delay measurements and known timing for the transmit (TX) and receive (RX) datapaths as required in applications such as wireless communication systems, emerging Ethernet standards, and test and measurement equipment. The Deterministic Latency PHY IP core support 1-32 lanes with a continuous range of data rates from 611 Mbps–11000 Mbps for Stratix V devices and 1–6.5536 Gbps for Arria V devices. By setting the appropriate options using the parameter editor, you can configure the Deterministic Latency PHY IP core to support many industry-standard protocols that require deterministic latency, including the following protocols:

- Common Public Radio Interface (CPRI)
- Open Base Station Architecture Initiative (OBSAI)
- 1588 Ethernet

Figure 9–1 illustrates the top-level interfaces and modules of the Deterministic Latency PHY IP core. As Figure 9–1, the physical coding sublayer (PCS) includes the following functions:

- TX and RX Phase Compensation FIFO
- Byte serializer and deserializer
- 8B/10B encoder and decoder
- Word aligner
- TX bit slipper

Figure 9–1. Deterministic Latency PHY IP Core



The data that the Deterministic Latency PHY receives data on its FPGA fabric interface employs the Avalon Streaming (Avalon-ST) protocol to transmit and receive data. The Avalon-ST protocol is a simple protocol designed for driving high bandwidth, low latency, unidirectional data. The Deterministic Latency PHY IP Core also includes an Avalon Memory-Mapped (Avalon-MM) interface to access control

and status registers. This is a standard, memory-mapped protocol that is normally used to read and write registers and memory. The transceiver reconfiguration interface connects to the Altera Transceiver Reconfiguration Controller IP core which can dynamically reconfigure transceiver settings. Finally, the PMA transmits and receives serial data which connects to an optical link.



For more information about the Avalon-ST and Avalon-MM protocols, refer to the [Avalon Interface Specifications](#).

Auto-Negotiation

The Deterministic Latency PHY IP core supports auto-negotiation. When auto-negotiation is required, the channels initialize at the highest supported frequency and switch to successively lower data rates if frame synchronization is not achieved. If your design requires auto-negotiation, choose a base data rate that minimizes the number of PLLs required to generate the clocks required for data transmission. By selecting an appropriate base data rate, you can change data rates by changing the divider used by the clock generation block. [Table 9-1](#) shows an example where setting two base data rates, 9830.4 and 6144 Mbps, with the appropriate clock dividers generates almost the full range of data rates required by the CPRI protocol.

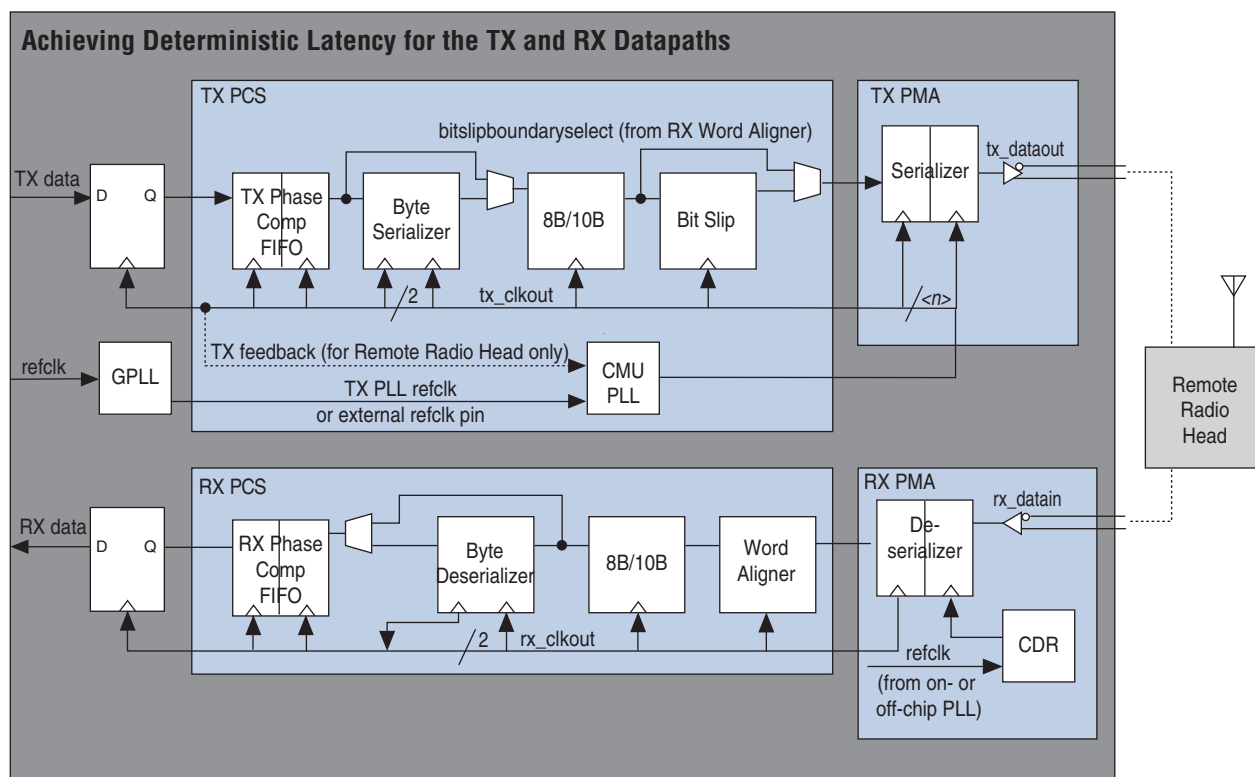
Table 9-1. Recommended Base Data Rate and Clock Divisors for CPRI

Data Rate (Mbps)	Base Data Rate (Mbps)	Clock Divider
614.4	4915.2	8
1228.8	4915.2	4
2457.6	4915.2	2
3072.0	6144.0	2
4915.2	4915.2	1
6144.0	6144.0	1

Achieving Deterministic Latency

Figure 9-2 illustrates the TX and RX channels when configured as a wireless basestation communicating to a remote radio head (RRH) using a CPRI or OBSAI interface. Figure 9-2 also provides an overview of the calculations that guarantee deterministic delay. As this figure illustrates, you can use a general-purpose PLL to generate the clock that drives the TX CMU PLL or an external reference clock input pin.

Figure 9-2. Achieving Deterministic Latency for the TX and RX Datapaths ⁽¹⁾



Note to Figure 9-2:

(1) The TX and RX Phase Compensation FIFOs always operate in register mode.

In the current release, Altera recommends that you use a delay estimate FIFO to measure the phase difference between the tx_clkout and rx_clkout, and the clock output of the PLL to ensure the delay through the FIFO to a certain accuracy.

Systems that require multiple PLLs in a single transceiver block must use a delay estimate FIFO to determine delay estimates and the required phase adjustments.

Delay Estimation Logic

This section provides the equations to calculate delays when the Deterministic Latency PHY IP core implements CPRI protocol. CPRI defines the radio base station interface between network radio equipment controllers (REC) and radio equipment (RE) components.

For RE

$$\begin{aligned} \text{RX\_latency\_RE} = & \langle \text{RX PCS latency in parallel clock cycles} \rangle \\ & + \langle \text{RX PMA latency in UI} \rangle \\ & + \langle \text{PMA uncertainty reported by wordalignment\_boundary}[5] \rangle \end{aligned}$$

$$\begin{aligned} \text{TX\_latency\_RE} = & \langle \text{TX PCS latency in parallel clock cycles} \rangle \\ & + \langle \text{TX PMA latency in UI} \rangle + \langle \text{Tx bitslip latency} \rangle \end{aligned}$$

For REC

$$\begin{aligned} \text{RX\_latency\_REC} = & \langle \text{RX PCS latency in parallel clock cycles} \rangle \\ & + \langle \text{RX PMA latency in UI} \rangle + \langle \text{rx\_clkout phase shift of tx\_clkout} \rangle \end{aligned}$$

$$\begin{aligned} \text{TX\_latency\_REC} = & \langle \text{TX PCS latency in parallel clock cycles} \rangle \\ & + \langle \text{TX PMA latency in UI} \rangle \end{aligned}$$

Round Trip Delay

$$\begin{aligned} \text{Launch\_time (from TX pins)} = & \langle \text{clock arrival time} \rangle + \langle \text{data arrival time} \rangle \\ = & \langle \text{clock arrival time} \rangle \\ & + \langle \text{TX latency in REC} \rangle \text{ (tx bitslip=0)} \\ = & \langle t_{PD} \text{ GPLL to CMU PLL} - t_{feedback} \rangle \\ & + ((\langle \text{TX\_latency in REC} \rangle \times \langle \text{tx\_clkout\_period} \rangle) \\ & + t_{TX\_tclock\_output}) \end{aligned}$$

$$\begin{aligned} \text{Arrival\_time (at RX pins)} = & \langle \text{latency time in RE} \rangle - \langle \text{RX latency time in REC} \rangle \\ = & (\langle \text{Round\_trip\_latency} \rangle \times \langle \text{tx\_clkout\_period} \rangle) \\ & - ((\langle \text{RX\_latency in REC} \rangle \times \langle \text{rx\_clkout\_period} \rangle) \\ & + \langle t_{PDIO>RX\_deser} \rangle \\ & + \langle \text{rx\_clkout\_phase\_WRT\_tx\_clkout}/360 \times \text{rx\_clkout\_period} \rangle) \end{aligned}$$

$$\text{Total Delay} = \langle \text{Arrival\_time} \rangle - \langle \text{Launch\_time} \rangle$$

Total Delay Uncertainty

Round trip delay estimates are subject to power, voltage, and temperature (PVT) variation.

$$t_{\text{RXCLK\_Phase\_detector\_uncertainty}} = 2 \times \max(\langle t_{\text{GLL\_phase\_step}} \rangle, \langle t_{\text{CDR\_to\_GPLL\_jitter}} \rangle) + \mu t_{\text{SU}} + \mu t_{\text{H}}$$

$$\begin{aligned} t_{\text{Round\_trip\_uncertainty}} = & \langle t_{\text{RX\_CLK\_Phase\_detector\_uncertainty}} \rangle + \langle t_{\text{GPLL} \rightarrow \text{CMU PLL\_variation}} \rangle \\ & + \langle t_{\text{feedback\_variation}} \rangle + \langle t_{\text{TX\_tco\_variation}} \rangle + \langle t_{\text{IO} \rightarrow \text{RXdeser\_delay\_variation}} \rangle \\ & + \langle t_{\text{PLL\_multicycle\_jitter}} \rangle + \langle t_{\text{offset\_uncertainty}} \rangle \end{aligned}$$

Delay Numbers

Table 9–2 shows the total latency through the TX PCS in parallel clock cycles. The TX compensation FIFO is in register mode.

Table 9–2. TX PCS Total Latency

Datapath Attributes	TX Phase Comp FIFO	Serializer	8B/10B	Bitslip	Total Clock Cycles
Single word with byte serializer	1.0	0.5	0.5	0	2.0
Single word without byte serializer	1.0	1.0	2.0	0	3.0

Table 9–2. TX PCS Total Latency

Datapath Attributes	TX Phase Comp FIFO	Serializer	8B/10B	Bitslip	Total Clock Cycles
Double word with byte serializer	1.0	0.5	0.5	0	2.0
Double word without byte serializer	1.0	1.0	2.0	0	3.0

Table 9–3 shows the total latency through the RX PCS in parallel clock cycles. The RX compensation FIFO is in register mode.

Table 9–3. RX PCS Total Latency

Datapath Attributes	RX Phase Comp FIFO	Byte Ordering	Deserializer	8B/10B	Word Aligner	Total Clock Cycles
Single word with byte deserializer	1.0	1.0	1.5	0.5	2.0	6.0
Single word without byte deserializer	1.0	1.0	1.0	1.0	4.0	8.0
Double word with byte deserializer	1.0	1.0	1.5	0.5	2.5	6.5
Double word without byte deserializer	1.0	1.0	1.0	1.0	5.0	9.0

Table 9–4 shows the total latency through the TX and RX PMA in UI.

Table 9–4. PMA Datapath Total Latency ⁽¹⁾

Device	TX PMA Latency in UI		RX PMA Latency in UI	
	Single Width with 10 bits	Double Width with 20 bits	Single Width with 10 bits	Double Width with 20 bits
Arria V	13	23	54	84
Stratix V	23	43	53	83

Note to Table 9–4:

(1) The numbers in this table are from simulation.



There is a small discrepancy between simulation of the delays through the PMA serializer and deserializer and hardware modeling of these delays.

Device Family Support

IP cores provide either final or preliminary support for target Altera device families. These terms have the following definitions:

- *Final support*—Verified with final timing models for this device.
- *Preliminary support*—Verified with preliminary timing models for this device.

Table 9–5 shows the level of support offered by the Deterministic Latency PHY IP core for Altera device families.

Table 9–5. Device Family Support

Device Family	Support
Arria V devices	Preliminary
Stratix V devices	Preliminary
Other device families	No support

Parameter Settings

To configure the Deterministic Latency PHY IP core in the parameter editor, click **Installed Plug-Ins > Interfaces > Transceiver PHY > Deterministic Latency PHY v11.1**. You can use the tabs on the parameter editor to select the options required for the protocol.

The following sections describe all of the options on the tabs of the parameter editor.

General Options

The **General Options** tab allows you to set the basic parameters of your transceiver PHY. [Table 9-6](#) lists the settings available on the **General Options** tab.

Table 9-6. General Options (Part 1 of 2)

Name	Value	Description
Device family	Arria V Stratix V	Specifies the device family. Arria V and Stratix V are available.
Mode of operation	Duplex TX RX	You can select to transmit data, receive data, or both.
Number of lanes	1–32	The total number of lanes in each direction.
FPGA fabric transceiver interface width	8 16 32	Specifies the word size between the FPGA fabric and PCS. Refer to “Sample Channel Width Options for Supported Serial Data Rates” on page 9-7 for the data rates supported at each word size.
PCS-PMA interface width	10 20	Specifies the datapath width between the transceiver PCS and PMA. A deserializer in the PMA receives serial input data from the RX buffer using the high-speed recovered clock and deserializes it using the low-speed parallel recovered clock.
PLL type	CMU ATX	Specifies the PLL type. The CMU PLL has a larger frequency range than the ATX PLL. The ATX PLL is designed to improve jitter performance and achieves lower channel-to-channel skew; however, it supports a narrower range of data rates and reference clock frequencies. Another advantage of the ATX PLL is that it does not use a transceiver channel, while the CMU PLL does. Because the CMU PLL is more versatile, it is specified as the default setting.
Data rate	611 Mbps – 6553.6 Mbps 611 Mbps - 11000 Mbps	The top data rate for Arria V is 6553.6 Mbps, for Stratix V it is 11000 Mbps. Refer to “Sample Channel Width Options for Supported Serial Data Rates” on page 9-7 for sample the channel widths that support these data rates.

Table 9-6. General Options (Part 2 of 2)

Name	Value	Description
Base data rate	$1 \times \text{Data rate}$ $2 \times \text{Data rate}$ $4 \times \text{Data rate}$ $8 \times \text{Data rate}$	For systems that transmit and receive data at more than one data rate, select a base data rate that minimizes the number of PLLs required to generate the clocks for data transmission. Table 9-1 on page 9-2 lists the recommended Base data rates for various Data rates. The available options are dynamically computed based on the Data rate you specified as long as those Base data rates are within the frequency range of the PLL.
Input clock frequency	$\text{Data rate}/20$ $\text{Data rate}/10$ $\text{Data rate}/8$ $\text{Data rate}/5$ $\text{Data rate}/4$ $\text{Data rate}/2.5$ $\text{Data rate}/2$ $\text{Data rate}/1.25$ $\text{Data rate}/1$	This is the reference clock for the PHY PLL. The available options are based on the Base data rate specified.

[Table 9-7](#) lists the available channel widths available at selected frequencies. The channel width options are restricted by the following maximum FPGA-PCS fabric interface frequencies:

- Arria V devices—159.375 MHz
- Stratix V devices—221 MHz when the byte serializer and deserializer are disabled
- Stratix V devices —325 MHz when the byte serializer and deserializer are enabled

Table 9-7. Sample Channel Width Options for Supported Serial Data Rates

Serial Data Rate (Mbps)	Channel Width (FPGA-PCS Fabric)			
	Single-Width		Double-Width	
	8-Bit	16-Bit	16-Bit	32-Bit
614.4	✓	✓	—	—
1228.8	✓	✓	✓	✓
2457.6	—	✓	✓	✓
3072	—	✓	✓	✓
4915.2	—	—	—	✓
6144	—	—	—	✓

Additional Options

Table 9–8 lists the settings available on the **Additional Options** tab.

Table 9–8. Additional Options (Part 1 of 2)

Name	Value	Description
Word alignment mode	Deterministic latency state machine	The word aligner restores word boundaries of received data based on a predefined alignment pattern. The word aligner automatically performs an initial alignment to the specified word pattern after reset deassertion. You can select 1 of the following 2 modes: ■ Deterministic latency state machine—In this mode, the RX word aligner automatically searches for the word alignment pattern after reset completes. After the word aligner detects the specified word alignment pattern, it sends <code>RX_CLKSLIP</code> to the RX PMA deserializer indicating the number of bits to slip to compensate for the bits that were slipped to achieve word alignment. When <code>RX_CLKSLIP</code> has a non-zero value, the deserializer either skips one serial bit or pauses the serial clock for one cycle. As a result, the period of the parallel clock could be extended by 1 unit interval (UI) during the clock slip operation. This procedure avoids using the TX bit slipper to ensure constant round-trip delay. In this mode, the specified word alignment pattern, which is currently forced to K28.5 (0011111010) is always placed in the least significant byte (LSB) of a word with a fixed latency of 3 cycles. User logic can assume the LSB placement. Altera recommends the deterministic latency state machine mode for new designs. During the word alignment process, the parallel clock shifts the phase to align to the data. This phase shifting will be 2/10 cycles (20%) in 10 bit mode, 2/20 cycles (10%) in 20 bit mode, and 2/40 cycles (5%) in 40 bit mode. For double-width datapaths using deterministic latency state machine mode, after the initial alignment following the deassertion of reset, the Avalon-MM register <code>big_rx_enapatternalign</code> (not available as a signal) must be reasserted to initiate another pattern alignment. Asserting <code>rx_enapatternalign</code>, may cause the extra shifting in the RX datapath if <code>rx_enablepatternalign</code> is asserted while bit slipping is in progress; consequently <code>rx_enapatternalign</code> should only be asserted under the following conditions: ■ <code>rx_syncstatus</code> is asserted ■ <code>rx_bitslipboundaryselectout</code> changes from a non-zero value to zero or 1
	Manual	■ Manual—In this mode, the RX word aligner parses the incoming data stream for a specific alignment character. Once it identifies this pattern, it shifts the input stream to align the data and also outputs the number of bits slipped on <code>bitslipboundaryselectot[4:0]</code> for latency compensation on the TX datapath. This mode is provided for backwards compatibility with designs implemented in Stratix IV and Arria II devices.

Table 9–8. Additional Options (Part 2 of 2)

Name	Value	Description
TX bitflip	On/Off	TX bitflip is enabled whenever the word aligner is in Manual alignment mode. The TX bitflipper uses the value of <code>bitflipboundarselect[4:0]</code> to compensate for bits slipped on the RX datapath to achieve deterministic latency.
Enable run length violation checking	On/Off	If you turn this option on, you can specify the run length which is the maximum legal number of contiguous 0s or 1s. This option also creates the <code>rx_rlv</code> output signal which is asserted when a run length violation is detected.
Run length	5–160	Specifies the threshold for a run-length violation. Must be a multiple of 5.
Create optional word aligner status ports	On/Off	Enable this option to include the <code>rx_patterndetect</code> and <code>rx_syncstatus</code> ports.
Create optional 8B/10B control and status ports	On/Off	Enable this option to include the 8B/10B <code>rx_runningdisp</code> , <code>rx_errdetect</code> , and <code>rx_disperserr</code> signals at the top level of the Deterministic Latency PHY IP core.
Create PMA optional status ports	On/Off	Enable this option to include the 8B/10B <code>rx_is_lockedtoref</code> , <code>rx_is_lockedtodata</code> , and <code>rx_signaldetect</code> signals at the top level of the Deterministic Latency PHY IP core.
Avalon data interfaces	On/Off	This option is typically required if you are planning to import your Deterministic Latency PHY IP core into a Qsys system.
Enable embedded reset controller	On/Off	When you turn this option On , the embedded reset controller handles reset of the TX and RX channels at power up. If you turn this option Off , you must design a reset controller that manages the following reset signals: <code>tx_digitalreset</code> , <code>tx_analogreset</code> , <code>tx_cal_busy</code> , <code>rx_digitalreset</code> , <code>rx_analogreset</code> , and <code>rx_cal_busy</code> .

Analog Options

You specify the analog parameters for Stratix V devices using the Quartus II Assignment Editor, the Pin Planner, or through the Quartus II Settings File (`.qsf`). The default values for analog options fall into three categories:

- *Global*—These parameters have default values that are independent of other parameter settings.
- *Computed*—These parameters have an initial default value that is recomputed based on other parameter settings.
- *Proxy*—These parameters have default values that are place holders. The Quartus II software selects these initial default values based on your design; however, Altera recommends that you replace these defaults with values that match your electrical board specification.

Table 9-9 lists the analog parameters for Stratix V devices whose original values are place holders for the values that match your electrical board specification. In Table 9-9, the default value of an analog parameter is shown in **bold** type.

Table 9-9. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 2)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_IO_PIN_TERMINATION	Transceiver I/O Pin Termination	Specifies the intended on-chip termination value for the specified transceiver pin. Use External Resistor if you intend to use off-chip termination.	85_OHMS 100_OHMS 120_OHMS 150_OHMS EXTERNAL_RESISTOR	Pin
XCVR_REFCLK_PIN_TERMINATION	Transceiver Dedicated Refclk Pin Termination	Specifies the intended termination value for the specified refclk pin.	DC_COUPLING_INTERNAL_100_OHM DC_COUPLING_EXTERNAL_RESISTOR AC_COUPLING	Pin
XCVR_RX_BYPASS_EQ_STAGES_234	Receiver Equalizer Stage 2, 3, 4 Bypass	Bypass continuous time equalizer stages 2, 3, and 4 to save power. This setting eliminates significant AC gain on the equalizer and is appropriate for chip-to-chip short range communication on a PCB.	ALL_STAGES_ENABLED BYPASS_STAGES	Pin
XCVR_TX_SLEW_RATE_CTRL	Transmitter Slew Rate Control	Specifies the slew rate of the output signal. The valid values span from the slowest rate to fastest rate with 1 representing the slowest rate.	1-5	Pin

Table 9–9. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 2)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_VCCA_VOLTAGE	VCCA_GXB Voltage	Configure the VCCA_GXB voltage for a GXB I/O pin by specifying the intended VCCA_GXB voltage for a GXB I/O pin. If you do not make this assignment the compiler automatically sets the correct VCCA_GXB voltage depending on the configured data rate, as follows: ■ Data rate <= 6.5 Gbps: 2_5V ■ Data rate > 6.5 Gbps: 3_0V or 3_3V for Stratix V ES silicon	2_5V 3_0V	Pin
XCVR_VCCR_VCCT_VOLTAGE	VCCR_GXB VCCT_GXB Voltage	Configure the VCCR_GXB and VCCT_GXB voltage for an GXB I/O pin by specifying the intended supply voltages for a GXB I/O pin. If your design uses decision feedback equalization (DFE) or adaptive equalization (AEQ), you must set this parameter to 1.0V. Otherwise, if you do not make this assignment the compiler automatically sets the correct VCCR_GXB and VCCT_GXB voltage depending on the configured data rate as follows: ■ Data rate <= 6.5 Gbps: 0_85V ■ Data rate > 6.5 Gbps: 1_0V	0_85V 1_0V	Pin

Table 9–10 lists the analog parameters with *global* or *computed* default values. You may want to optimize some of these settings. In Table 9–10, the default value is shown in **bold** type. For computed analog parameters, the default value listed is for the initial setting, not the recomputed setting.

Table 9–10. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 3)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
Analog Parameters with Global Default Value				
CDR_BANDWIDTH_PRESET	CDR Bandwidth Preset	Specifies the CDR bandwidth preset setting.	Auto Low Medium High	PLL instance
PLL_BANDWIDTH_PRESET	PLL Bandwidth Preset	Specifies the PLL bandwidth preset setting	Auto Low Medium High	PLL instance
XCVR_RX_DC_GAIN	Receiver Buffer DC Gain Control	Controls the amount of a stage receive-buffer DC gain.	0–4	Pin

Table 9–10. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 3)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_RX_LINEAR_EQUALIZER_CONTROL	Receiver Linear Equalizer Control	Static control for the continuous time equalizer in the receiver buffer. The equalizer has 16 distinct settings from 0–15 corresponding to the increasing AC gain.	1–16	Pin
Analog Parameter with Computed Default Value				
XCVR_RX_COMMON_MODE_VOLTAGE	Receiver Buffer Common Mode Voltage	Receiver buffer common-mode voltage.	VTT_OP80V VTT_OP75V VTT_OP70V VTT_OP65V VTT_OP60V VTT_OP55V VTT_OP50V VTT_OP35V VTT_PUP_WEAK VTT_PDN_WEAK TRISTATE1 VTT_PDN_STRONG VTT_PUP_STRONG TRISTATE2 TRISTATE3 TRISTATE4	Pin
XCVR_RX_ENABLE_LINEAR_EQUALIZER_PCIEMODE	Receiver Linear Equalizer Control (PCI Express)	If enabled equalizer gain control is driven by the PCS block for PCI Express. If disabled equalizer gain control is determined by the XCVR_RX_LINEAR_EQUALIZER_SETTING assignment.	TRUE FALSE	Pin
XCVR_RX_EQ_BW_SEL	Receiver Equalizer Gain Bandwidth Select	Sets the gain peaking frequency for the equalizer. For data-rates of less than 6.5Gbps set to HALF. For higher data-rates set to FULL.	FULL HALF	Pin
XCVR_RX_SD_ENABLE	Receiver Signal Detection Unit Enable/Disable	Enables or disables the receiver signal detection unit.	TRUE FALSE	Pin
XCVR_RX_SD_OFF	Receiver Cycle Count Before Signal Detect Block Declares Loss Of Signal	Number of parallel cycles to wait before the signal detect block declares loss of signal.	0 –29	Pin
XCVR_RX_SD_ON	Receiver Cycle Count Before Signal Detect Block Declares Presence Of Signal	Number of parallel cycles to wait before the signal detect block declares presence of signal.	0 –16	Pin
XCVR_RX_SD_THRESHOLD	Receiver Signal Detection Voltage Threshold	Specifies signal detection voltage threshold level.	0 –7	Pin

Table 9-10. Transceiver and PLL Assignments for Stratix V Devices (Sheet 1 of 3)

QSF Assignment Name	Pin Planner and Assignment Editor Name	Description	Options	Assign To
XCVR_TX_COMMON_MODE_VOLTAGE	Transmitter Common Mode Driver Voltage	Transmitter common-mode driver voltage	VOLT_OP80V VOLT_OP75V VOLT_OP70V VOLT_OP65V VOLT_OP60V VOLT_OP55V VOLT_OP50V VOLT_OP35V PULL_UP PULL_DOWN TRISTATED1 GROUNDED PULL_UP_TO_VCCELA TRISTATED2 TRISTATED3 TRISTATED4	Pin
XCVR_TX_PRE_EMP_1ST_POST_TAP	Transmitter Preemphasis First Post-Tap	Specifies the first post-tap setting value.	0 –31	Pin
XCVR_TX_PRE_EMP_2ND_POST_TAP	Transmitter Preemphasis Second Post-Tap	Specifies the second post-tap setting value.	0 –15	Pin
XCVR_TX_PRE_EMP_INV_2ND_TAP	Transmitter Preemphasis Second Tap Invert	Inverts the transmitter pre-emphasis 2nd post tap.	TRUE FALSE	Pin
XCVR_TX_PRE_EMP_INV_PRE_TAP	Transmitter Preemphasis Pre Tap Invert	Inverts the transmitter pre-emphasis pre-tap.	TRUE FALSE	Pin
XCVR_TX_PRE_EMP_PRE_TAP	Transmitter Preemphasis Pre-Tap	Specifies the pre-tap pre-emphasis setting.	0 –15	Pin
XCVR_TX_RX_DET_ENABLE	Transmitter's Receiver Detect Block Enable	Enables or disables the receiver detector circuit at the transmitter.	TRUE FALSE	Pin
XCVR_TX_RX_DET_MODE	Transmitter's Receiver Detect Block Mode	Sets the mode for receiver detect block	0 –15	Pin
XCVR_TX_RX_DET_OUTPUT_SEL	Transmitter's Receiver Detect Block QPI/PCI Express Control	Determines QPI or PCI Express mode for the Receiver Detect block.	RX_DET_QPI_OUT RX_DET_PCIE_OUT	Pin
XCVR_TX_VOD	Transmitter Differential Output Voltage	Differential output voltage setting. The values are monotonically increasing with the driver main tap current strength.	0–63 50	Pin
XCVR_TX_VOD_PRE_EMP_CTRL_SRC	Transmitter V _{OD} /Preemphasis Control Source	When set to DYNAMIC_CTL, the PCS block controls the V _{OD} and preemphasis coefficients for PCI Express. When this assignment is set to RAM_CTL the V _{OD} and preemphasis are controlled by other assignments, such as XCVR_TX_PRE_EMP_1ST_POST_TAP.	DYNAMIC_CTL RAM_CTL	Pin

❓ For more information about the Pin Planner, refer to [About the Pin Planner](#) in Quartus II Help. For more information about the Assignment Editor, refer to [About the Assignment Editor](#) in Quartus II Help.

🔧 For more information about Quartus II Settings, refer to [Quartus II Settings File Manual](#).

Interfaces

This section describes interfaces of the Deterministic Latency Transceiver PHY. It includes the following topics:

- [Ports](#)
- [Register Interface](#)
- [Dynamic Reconfiguration](#)

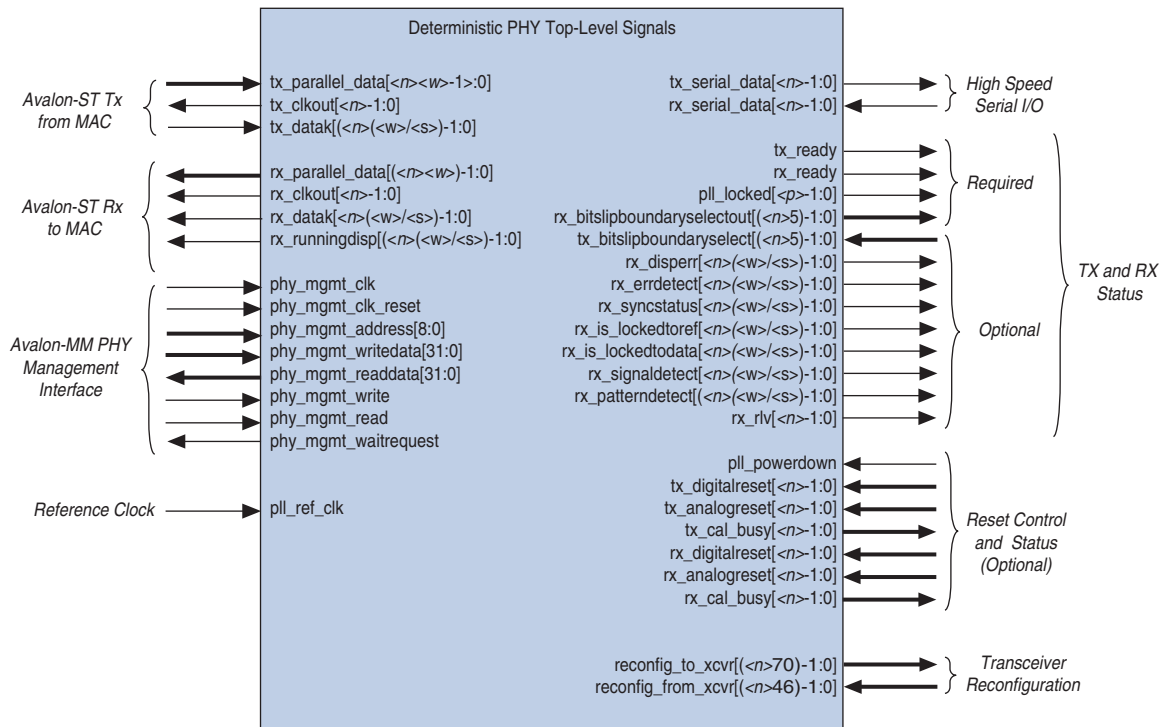
Ports

[Figure 9-3](#) illustrates the top-level signals of the Deterministic Latency PHY IP core. The variables in [Figure 9-3](#) represent the following parameters:

- $\langle n \rangle$ —The number of lanes
- $\langle w \rangle$ —The width of the FPGA fabric to transceiver interface per lane
- $\langle s \rangle$ — The symbol size

■ $\langle p \rangle$ —The number of PLLs

Figure 9-3. Deterministic Latency PHY Top-Level Signals



The **block diagram** shown in the parameter editor labels the external pins with the *interface type* and places the *interface name* inside the box. The interface type and name are used in the `_hw.tcl` file that describes the component. If you turn on **Show signals**, the **block diagram** displays all top-level signal names.

For more information about `_hw.tcl` files refer to the [Component Interface Tcl Reference](#) chapter in volume 1 of the *Quartus II Handbook*.

The following sections describe the signals in each interface.

Avalon-ST TX Input Data from the MAC

Table 9–11 describes the signals in the Avalon-ST input interface. These signals are driven from the MAC to the PCS. This is an Avalon sink interface.



For more information about the Avalon-ST protocol, including timing diagrams, refer to the *Avalon Interface Specifications*.

Table 9–11. Avalon-ST TX Interface

Signal Name	Direction	Description
tx_parallel_data[(<n><w>)-1:0]	Sink	This is TX parallel data driven from the MAC. The ready latency on this interface is 0, so that the PHY must be able to accept data as soon as it comes out of reset.
tx_clkout[<n>-1:0]	Output	This is the clock for TX parallel data, control, and status signals.
tx_dataak[(<n>(<d>/<s>)-1:0]	Sink	Data and control indicator for the received data. When 0, indicates that tx_parallel_data is data, when 1, indicates that tx_parallel_data is control.

Avalon-ST RX Output Data to the MAC

Table 9–12 describes the signals in the Avalon-ST output interface. These signals are driven from the PCS to the MAC. This is an Avalon source interface.

Table 9–12. Avalon-ST RX Interface

Signal Name	Direction	Description
rx_parallel_data[(<n><d>)-1:0]	Source	This is RX parallel data driven from the Deterministic Latency PHY IP core. The ready latency on this interface is 0, so that the MAC must be able to accept data as soon as the PHY comes out of reset. Data driven from this interface is always valid.
rx_clkout[<n>-1:0]	Output	This is the clock for the RX parallel data source interface.
rx_dataak[(<n>(<d>/<s>)-1:0]	Source	Data and control indicator for the source data. When 0, indicates that rx_parallel_data is data, when 1, indicates that rx_parallel_data is control.
rx_runningdisp[(<n>(<d>/<s>)-1:0]	Source	This status signal indicates the disparity of the incoming data.

Clock Interface

Table 9–13 describes clocks for the Deterministic Latency PHY. The input reference clock, pll_ref_clk, drives a PLL inside the PHY-layer block, and a PLL output clock, rx_clkout, is used for all data, command, and status inputs and outputs.

Table 9–13. Clock Signals

Signal Name	Direction	Description
pll_ref_clk	Input	Reference clock for the PHY PLLs. Frequency range is 60–700 MHz.

Transceiver Serial Data Interface

Table 9-14 describes the differential serial data interface and the status signals for the transceiver serial data interface.

Table 9-14. Serial Interface and Status Signals ⁽¹⁾

Signal Name	Direction	Signal Name
rx_serial_data[<n>-1:0]	Input	Receiver differential serial input data.
tx_serial_data[<n>-1:0]	Output	Transmitter differential serial output data.

Note to Table 9-14:

(1) <n> is the number of lanes.

TX and RX Status Signals

Table 9-15 describes the optional status signals for the RX interface.

Table 9-15. Serial Interface and Status Signals (Part 1 of 2) ⁽¹⁾

Signal Name	Direction	Signal Name
tx_ready	Output	When asserted, indicates that the TX interface has exited the reset state and is ready to transmit.
rx_ready	Output	When asserted, indicates that the RX interface has exited the reset state and is ready to receive.
pll_locked[<p>-1:0]	Output	When asserted, indicates that the PLL is locked to the input reference clock.
rx_bitslipboundaryselectout [(<n>5)-1:0]	Output	Specifies the number of bits slipped to achieve word alignment. In 3G (10-bit) mode, the output is the number of bits slipped. If no bits were slipped, the output is 0. In 6G (20-bit) mode, the output is (19 – the number of bits slipped). If no bits were slipped, the output is 19. The default value of rx_bitslipboundaryselectout[4:0] before alignment is achieved is 5'b01111 in 3G mode and 5'b11111 in 6G mode.
Optional Status Signals		
tx_bitslipboundaryselect [(<n>5)-1:0]	Input	This signal is used for bit slip word alignment mode. It selects the number of bits that the TX block must slip to achieve a deterministic latency.
rx_disperr[(<n>(<d>/<s>)-1:0]	Output	When asserted, indicates that the received 10-bit code or data group has a disparity error.
rx_errdetect[(<n>(<d>/<s>)-1:0]	Output	When asserted, indicates that a received 10-bit code group has an 8B/10B code violation or disparity error.
rx_syncstatus[(<n>(<d>/<s>)-1:0]	Output	Indicates presence or absence of synchronization on the RX interface. Asserted when word aligner identifies the word alignment pattern or synchronization code groups in the received data stream. This signal is optional.
rx_is_lockedtoref[(<n>(<d>/<s>)-1:0]	Output	Asserted when the receiver CDR is locked to the input reference clock. This signal is asynchronous. This signal is optional.

Table 9–15. Serial Interface and Status Signals (Part 2 of 2)⁽¹⁾

Signal Name	Direction	Signal Name
<code>rx_is_lockedtoata[(<n>(<d>/<s>)-1:0]</code>	Output	When asserted, the receiver CDR is in to lock-to-data mode. When deasserted, the receiver CDR lock mode depends on the <code>rx_locktoarefclk</code> signal level. This signal is optional.
<code>rx_patterndetect[(<n>(<d>/<s>)-1:0]</code>	Output	When asserted, indicates that the programmed word alignment pattern has been detected in the current word boundary.
<code>rx_rlv[<n>-1:0]</code>	Output	When asserted, indicates a run length violation. Asserted if the number of consecutive 1s or 0s exceeds the number specified using the parameter editor.

Optional Reset Control and Status

Table 9–16 describes the signals in the optional reset control and status interface. These signals are available if you do not enable the embedded reset controller. For more information including timing diagrams, refer to *Transceiver Reset Control in Stratix V Devices* in volume 3 of the *Stratix V Device Handbook*.

Table 9–16. Avalon-ST RX Interface

Signal Name	Direction	Description
<code>pll_powerdown[<n>-1:0]</code>	Input	When asserted, resets the TX PLL.
<code>tx_digitalreset[<n>-1:0]</code>	Input	When asserted, reset all blocks in the TX PCS.
<code>tx_analogreset[<n>-1:0]</code>	Input	When asserted, resets all blocks in the TX PMA.
<code>tx_cal_busy[<n>-1:0]</code>	Output	When asserted, indicates that the TX channel is being calibrated. You must hold the channel in reset until calibration completes.
<code>rx_digitalreset[<n>-1:0]</code>	Input	When asserted, resets the RX PCS.
<code>rx_analogreset[<n>-1:0]</code>	Input	When asserted, resets the RX CDR.
<code>rx_cal_busy[<n>-1:0]</code>	Output	When asserted, indicates that the RX channel is being calibrated. You must hold the channel in reset until calibration completes.

Register Interface

The Avalon-MM PHY management interface provides access to the Deterministic Latency PHY PCS and PMA registers that control the TX and RX channels, the PMA powerdown and PLL registers, and loopback modes.

Figure 9-4 illustrates the role of the PHY Management module in the Deterministic Latency PHY.

Figure 9-4. Deterministic Latency PHY IP Core

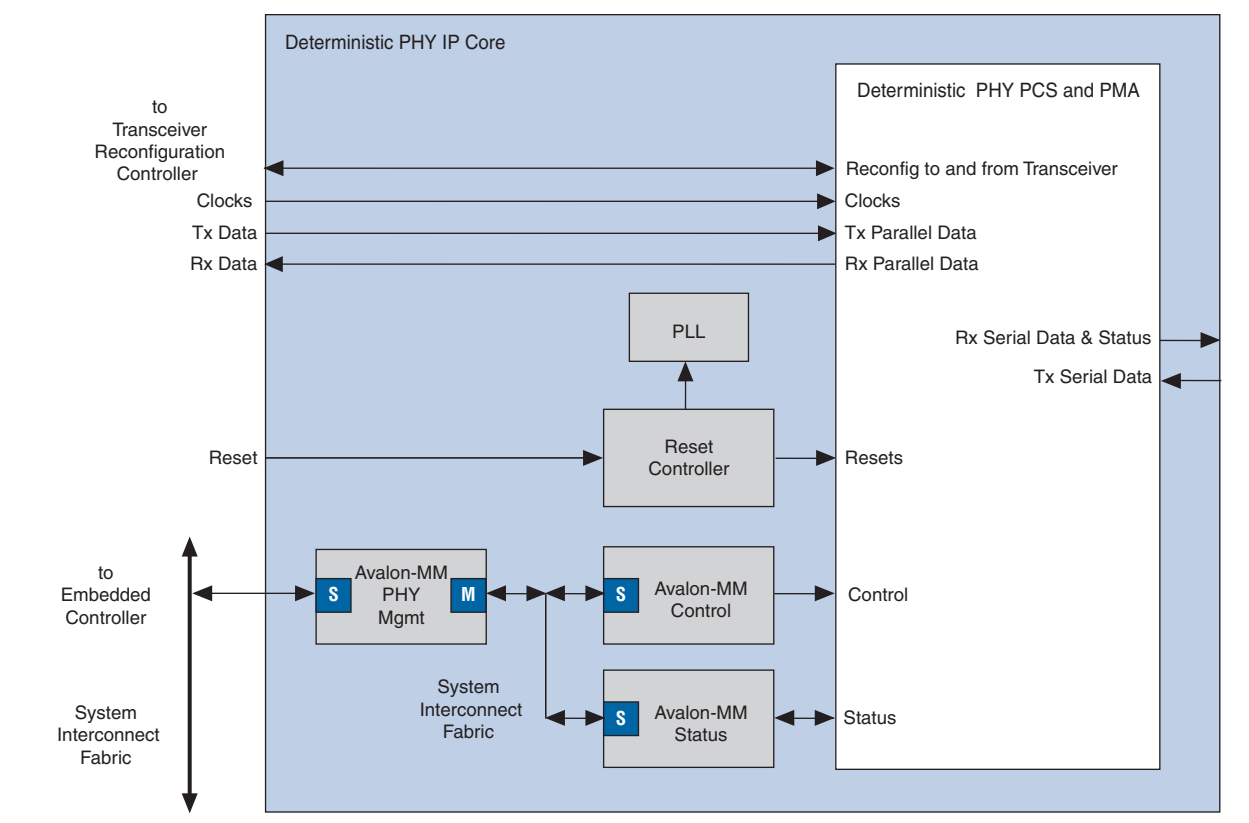


Table 9-17 describes the signals in the PHY Management interface.

Table 9-17. Avalon-MM PHY Management Interface (Part 1 of 2)

Signal Name	Direction	Description
phy_mgmt_clk	Input	Avalon-MM clock input. There is no frequency restriction for Stratix V devices; however, if you plan to use the same clock for the PHY management interface and transceiver reconfiguration, you must restrict the frequency range of phy_mgmt_clk to 100–125 MHz to meet the specification for the transceiver reconfiguration clock.
phy_mgmt_clk_reset	Input	Global reset signal. This signal is active high and level sensitive.
phy_mgmt_address[8:0]	Input	9-bit Avalon-MM address.
phy_mgmt_writedata[31:0]	Input	Input data.
phy_mgmt_readdata[31:0]	Output	Output data.
phy_mgmt_write	Input	Write signal.

Table 9–17. Avalon-MM PHY Management Interface (Part 2 of 2)

Signal Name	Direction	Description
phy_mgmt_read	Input	Read signal.
phy_mgmt_waitrequest	Output	When asserted, indicates that the Avalon-MM slave interface is unable to respond to a read or write request. When asserted, control signals to the Avalon-MM slave interface must remain constant.

Register Descriptions

Table 9–18 specifies the registers that you can access over the PHY management interface using word addresses and a 32-bit embedded processor. A single address space provides access to all registers.



Writing to reserved or undefined register addresses may have undefined side effects.

Table 9–18. Deterministic Latency PHY IP Core Registers (Part 1 of 3)

Word Addr	Bits	R/W	Register Name	Description
PMA Common Control and Status Registers				
0x021	[31:0]	RW	cal_blk_powerdown	Writing a 1 to channel <n> powers down the calibration block for channel <n>.
0x022	[31:0]	R	pma_tx_pll_is_locked	Bit[P] indicates that the TX CMU PLL (P) is locked to the input reference clock. There is typically one pma_tx_pll_is_locked bit per system.
Reset Control Registers—Automatic Reset Controller				
0x041	[31:0]	RW	reset_ch_bitmask	Reset controller channel bitmask for digital resets. The default value is all 1s. Channel <n> can be reset when bit<n> = 1.
0x042	[1:0]	W	reset_control (write)	Writing a 1 to bit 0 initiates a TX digital reset using the reset controller module. The reset affects channels enabled in the reset_ch_bitmask. Writing a 1 to bit 1 initiates a RX digital reset of channels enabled in the reset_ch_bitmask.
		R	reset_status(read)	Reading bit 0 returns the status of the reset controller TX ready bit. Reading bit 1 returns the status of the reset controller RX ready bit.

Table 9–18. Deterministic Latency PHY IP Core Registers (Part 2 of 3)

Word Addr	Bits	R/W	Register Name	Description
Reset Controls –Manual Mode				
0x044	[31:0]	RW	reset_fine_control	You can use the reset_fine_control register to create your own reset sequence. In manual mode, only the TX reset occurs automatically at power on and when the phy_mgmt_clk_reset is asserted. When pma_rx_setlocktodata or pma_rx_setlocktoref is set, the transceiver PHY is placed in manual mode.
	[31:4,0]	RW	Reserved	It is safe to write 0s to reserved bits.
	[3]	RW	reset_rx_digital	Writing a 1 causes the internal RX digital reset signal to be asserted, resetting the RX digital channels enabled in reset_ch_bitmask. You must write a 0 to clear the reset condition.
	[2]	RW	reset_rx_analog	Writing a 1 causes the internal RX analog reset signal to be asserted, resetting the RX analog logic of all channels enabled in reset_ch_bitmask. You must write a 0 to clear the reset condition.
	[1]	RW	reset_tx_digital	Writing a 1 causes the internal TX digital reset signal to be asserted, resetting all channels enabled in reset_ch_bitmask. You must write a 0 to clear the reset condition.
PMA Control and Status Registers				
0x061	[31:0]	RW	phy_serial_loopback	Writing a 1 to channel <n> puts channel <n> in serial loopback mode. For information about pre- or post-CDRserial loopback modes, refer to “ Loopback Modes ” on page 10–39 .
0x064	[31:0]	RW	pma_rx_set_locktodata	When set, programs the RX CDR PLL to lock to the incoming data. Bit <n> corresponds to channel <n>.
0x065	[31:0]	RW	pma_rx_set_locktoref	When set, programs the RX CDR PLL to lock to the reference clock. Bit <n> corresponds to channel <n>.
0x066	[31:0]	R	pma_rx_is_lockedtodata	When asserted, indicates that the RX CDR PLL is locked to the RX data, and that the RX CDR has changed from LTR to LTD mode. Bit <n> corresponds to channel <n>.
0x067	[31:0]	R	pma_rx_is_lockedtoref	When asserted, indicates that the RX CDR PLL is locked to the reference clock. Bit <n> corresponds to channel <n>.
PCS				
0x080	[31:0]	RW	Lane or group number	Specifies lane or group number for indirect addressing, which is used for all PCS control and status registers. For variants that stripe data across multiple lanes, this is the logical group number. For non-bonded applications, this is the logical lane number.
0x081	[31:6]	R	pcs8g_rx_status	Reserved.
	[5:1]	R	rx_bitslipboundaryselect out	This is an output from the bit slip word aligner which shows the number of bits slipped. From block: Word aligner.
	[0]	R	Reserved	—

Table 9–18. Deterministic Latency PHY IP Core Registers (Part 3 of 3)

Word Addr	Bits	R/W	Register Name	Description
0x082	[31:1]	R	pcs8g_tx_status	Reserved.
	[0]	RW	Reserved	—
0x083	[31:6]	RW	pcs8g_tx_control	Reserved.
	[5:1]	RW	tx_bitslipboundary_select	Sets the number of bits that the TX bit slipper needs to slip. To block: Word aligner.
	[0]	RW	tx_invpolarity	When set, the TX interface inverts the polarity of the TX data. To block: 8B/10B encoder.
0x084	[31:1]	RW	Reserved.	—
	[0]	RW	rx_invpolarity	When set, the RX channels inverts the polarity of the received data. To block: 8B/10B decoder.
0x085	[31:4]	RW	pcs8g_rx_wa_control	Reserved.
	[3]	RW	rx_bitslip	Every time this register transitions from 0 to 1, the RX data slips a single bit. To block: Word aligner.
	[2]	RW	rx_bytereversal_enable	When set, enables byte reversal on the RX interface. To block: Byte deserializer.
	[1]	RW	rx_bitreversal_enable	When set, enables bit reversal on the RX interface. To block: Word aligner.
	[0]	RW	rx_enapatternalign	When set in manual word alignment mode, the word alignment logic begins operation when this bit is set. To block: Word aligner.

Dynamic Reconfiguration

As silicon progresses towards smaller process nodes, circuit performance is affected more by variations due to process, voltage, and temperature (PVT). These process variations result in analog voltages that can be offset from required ranges. The calibration performed by the dynamic reconfiguration interface compensates for variations due to PVT.

Each channel and each TX PLL has a separate dynamic reconfiguration interfaces. The parameter editor provides informational messages on the connectivity of these interfaces. [Example 9–1](#) shows the messages for a single duplex channel.

Although you must initially create a separate reconfiguration interface for each channel and TX PLL in your design, when the Quartus II software compiles your design, it reduces the number of reconfiguration interfaces by merging reconfiguration interfaces. The synthesized design typically includes a reconfiguration interface for three channels. Allowing the Quartus II software to merge reconfiguration interfaces gives the Fitter more flexibility in placing transceiver channels.

Example 9-1. Informational Messages for the Transceiver Reconfiguration Interface

PHY IP will require 2 reconfiguration interfaces for connection to the external reconfiguration controller.
Reconfiguration interface offset 0 is connected to the transceiver channel.
Reconfiguration interface offset 1 is connected to the transmit PLL.

Table 9-19 describes the signals in the reconfiguration interface. This interface uses the Avalon-MM PHY Management interface clock.

Table 9-19. Reconfiguration Interface

Signal Name	Direction	Description
reconfig_to_xcvr [(<n>70)-1:0]	Sink	Reconfiguration signals from the Transceiver Reconfiguration Controller. <n> grows linearly with the number of reconfiguration interfaces.
reconfig_from_xcvr [(<n>46)-1:0]	Source	Reconfiguration signals to the Transceiver Reconfiguration Controller. <n> grows linearly with the number of reconfiguration interfaces.

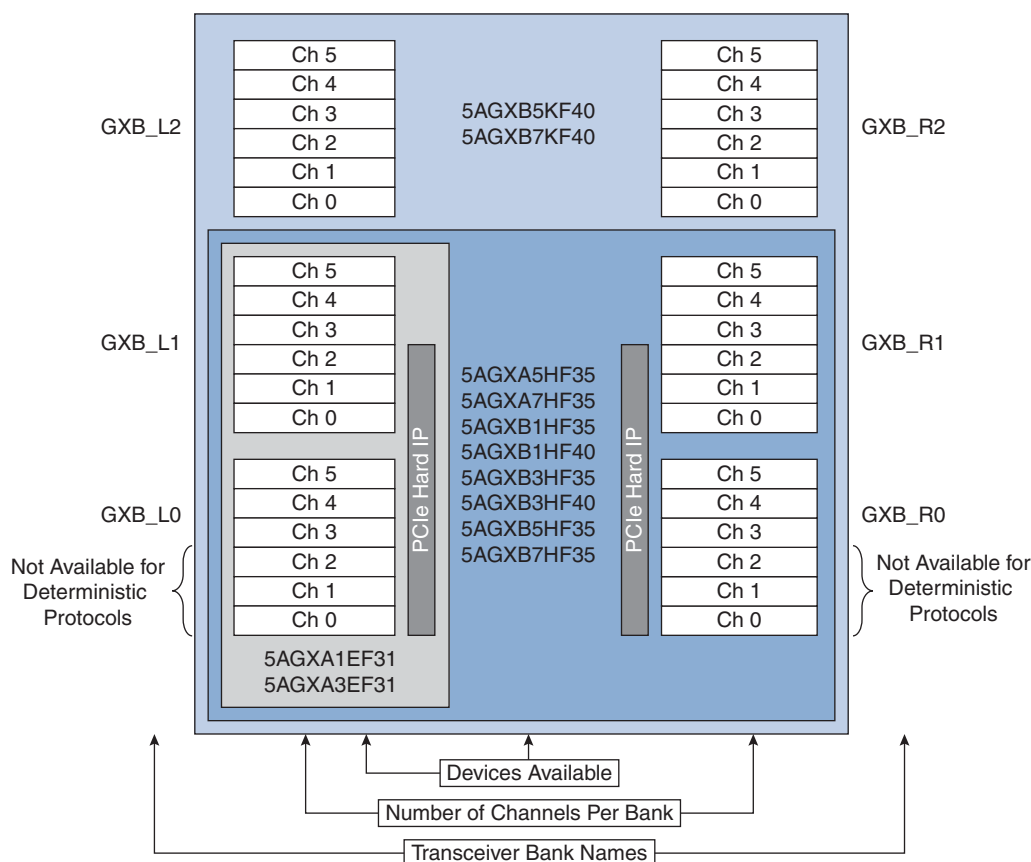
Channel Placement and Utilization

The Deterministic Latency PHY IP core has the following restriction on channel placement:

- Channels 0-2 in transceiver banks GXB_L0 and GSB_R0 of Arria V devices are not available for deterministic latency protocols.

Figure 9–5 shows the placement of transceiver banks in Arria V devices and indicates the channels that are not available.

Figure 9–5. Channel Placement and Available Channels in Arria V Devices



Simulation Files and Example Testbench

Refer to “Running a Simulation Testbench” on page 1–4 for a description of the directories and files that the Quartus II software creates automatically when you generate your Deterministic Latency PHY IP core.

This chapter provides additional information about the document and Altera.

The Altera Transceiver Reconfiguration Controller dynamically reconfigures the analog, channel, and TX PLL settings in Stratix V GX devices. Because the Stratix V FPGA is a 28-nm device, circuit performance is affected by variations due to process, voltage, and temperature (PVT). Stratix V devices that include transceivers must use the Transceiver Reconfiguration Controller to compensate for these variations. You can also use the Transceiver Reconfiguration Controller to compensate for signal integrity issues on the PCB or to change channel and TX PLL settings. For example, you can reconfigure a channel to support Ethernet running at 1 Gbps or 10 Gbps. You can also change the width of the datapath. The Transceiver Reconfiguration Controller provides access to the following settings:

- Transceiver calibration functions
- ATX PLL calibration
- PMA analog controls
- Adaptive equalization (AEQ)

The Transceiver Reconfiguration Controller also provides access to the following channel and TX PLL settings:

- The reference clock input to the TX PLL
- The clock dividers used by the TX PLL
- The PCS datapath settings
- Loopback modes: both pre-CDR (clock data recovery) reverse serial loopback and post-CDR reverse serial loopback are supported



In the Quartus II 11.1 release, the Transceiver Reconfiguration Controller is available for Arria V devices. However, it is not functional. It is available so that you can include it in your design to establish the connections necessary between the two IP cores.

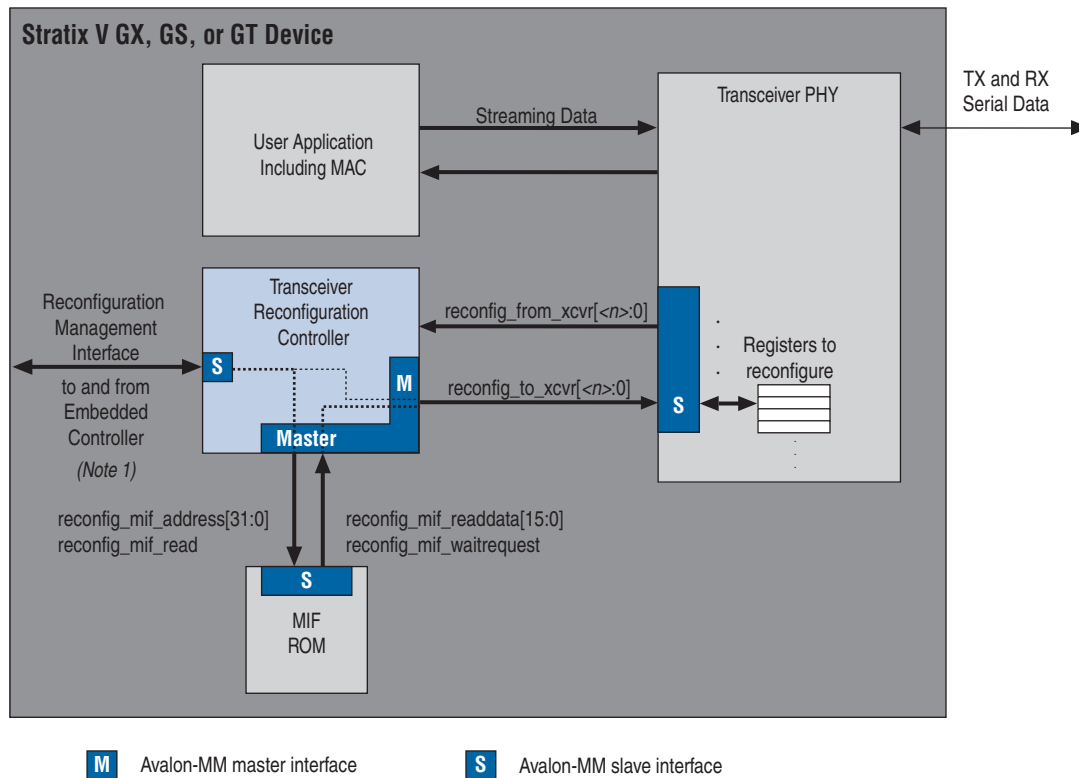
This user guide describes the features of the Transceiver Reconfiguration Controller. It also includes descriptions of the accessible transceiver registers, information about the MIF file format, and examples demonstrating the update procedures. It includes the following sections:

- “System Overview” on page 10-3
- “Device Family Support” on page 10-4
- “Performance and Resource Utilization” on page 10-5
- “Parameter Settings” on page 10-5
- “Interfaces” on page 10-7
- “Reconfiguration Controller Memory Map” on page 10-9
- “PMA Analog Controls” on page 10-11
- “EyeQ” on page 10-12
- “DFE” on page 10-14
- “AEQ” on page 10-16
- “ATX PLL Calibration” on page 10-17
- “PLL Reconfiguration” on page 10-18
- “Channel and PLL Reconfiguration” on page 10-21
- “Streamer Module” on page 10-22
- “Procedures for Reconfiguration” on page 10-27
- “Understanding Logical Channel Numbering” on page 10-31
- “Reconfiguration Controller to PHY IP Connectivity” on page 10-37
- “Merging TX PLLs In Multiple Transceiver PHY Instances” on page 10-38
- “Loopback Modes” on page 10-39

System Overview

Figure 10-1 illustrates the Transceiver Reconfiguration Controller's role in a Stratix V design.

Figure 10-1. Transceiver Reconfiguration Controller



Note to Figure 10-1:

(1) You can locate the embedded controller on-chip or on the PCB.

As Figure 10-1 illustrates, an embedded controller programs the Transceiver Reconfiguration Controller using its Avalon-MM slave interface. The `reconfig_to_xcvr` and `reconfig_from_xcvr` buses include the Avalon-MM address, read, write, readdata, writedata, and signals that connect to features related to calibration and signal integrity.

The Transceiver Reconfiguration Controller provides two modes to dynamically reconfigure transceiver settings:

- **Register Based**—In this access mode you can directly reconfigure a transceiver PHY IP core using the Transceiver Reconfiguration Controller's reconfiguration management interface. You initiate reconfiguration using a series of Avalon-MM reads and writes to the appropriate registers of the Transceiver Reconfiguration Controller. The Transceiver Reconfiguration Controller translates the device independent commands received on the reconfiguration management interface to device dependent commands on the transceiver reconfiguration interface. For more information, refer to "Channel and PLL Reconfiguration" on page 10-21.



For more information about Avalon-MM interfaces including timing diagrams, refer to the [Avalon Interface Specifications](#).

- **Streamer Based**—This access mode allows you to either stream a MIF that contains the reconfiguration data or perform direct writes to perform reconfiguration. The streaming mode uses a memory initialization file (**.mif**) to stream an update to the transceiver PHY IP core. The **.mif** file can contain changes for many settings. For example, a single **.mif** file might contain changes to the PCS datapath settings, clock settings, and PLL parameters. You specify the **.mif** using write commands on the Avalon-MM PHY management interface. After the streaming operation is specified, the update proceeds in a single step. For more information, refer to “[Streamer Based Reconfiguration](#)” on page 10-28. In the direct write mode, you perform Avalon-MM reads and writes to initiate a reconfiguration of the PHY IP. For more information, refer to “[Direct Write Reconfiguration](#)” on page 10-29.

Table 10-1 shows the features that you can reconfigure or control using register-based and MIF-based access modes.

Table 10-1. Reconfiguration Feature Access Modes

Feature	Register-Based	Streamer-Based
PMA settings, including V_{OD} , pre-emphasis, RX equalization DC gain, RX equalization control	✓	✓
Pre-CDR and post-CDR loopback modes	✓	—
AEQ mode	✓	—
Eye Monitor	✓	—
ATX Tuning	✓	—
Reference clock	—	✓
TX PLL clock switching	✓	✓
Channel interface	—	✓
Channel internals	—	✓

Device Family Support

IP cores provide either final or preliminary support for target Altera device families. These terms have the following definitions:

- *Final support*—Verified with final timing models for this device.
- *Preliminary support*—Verified with preliminary timing models for this device.

Table 10-2 lists the level of support offered by the Transceiver Reconfiguration Controller for Altera device families.

Table 10-2. Device Family Support

Device Family	Support
Stratix V devices	Preliminary
Other device families	No support

Performance and Resource Utilization

Table 10–3 shows the approximate device resource utilization for a the Transceiver Reconfiguration Controller. The numbers of combinational ALUTs and logic registers are rounded to the nearest 50. Table 10–3 also shows the time required for calibration and AEQ functions.

Table 10–3. Resource Utilization

Component	ALUTs	Registers	Memory Blocks	M20Ks	Run Time
Transceiver Calibration Functions					
Offset Cancellation	500	400	0	0	100 μ s/channel
Duty cycle calibration	350	400	0	0	70 μ s/channel
ATX PLL calibration	650	450	0	4	60 μ s/channel
Analog Features					
EyeQ	300	200	0	0	—
AEQ	700	500	0	0	40 μ s/channel
Reconfiguration Features					
Channel and PLL reconfiguration	400	500	0	0	— ⁽¹⁾
PLL reconfiguration (only)	250	350	0	0	— ⁽¹⁾

Note to Table 10–3:

(1) The time to complete these functions depends upon the complexity of the reconfiguration operation.

Parameter Settings

The Transceiver Reconfiguration Controller is available in the MegaWizard Plug-In Manager and Qsys design flows. To configure the Transceiver Reconfiguration Controller IP core in the MegaWizard design flow, click **Installed Plug-Ins > Interfaces > Transceiver PHY > Transceiver Reconfiguration Controller v11.1**. To configure the Transceiver Reconfiguration Controller in Qsys, in the **Component Library**, type Transc in the **Search Box**. Qsys filters all available components for this text string and displays the Transceiver Reconfiguration Controller which is in the **Interface Protocols > Transceiver PHY** category.

Table 10-4 lists the available options.

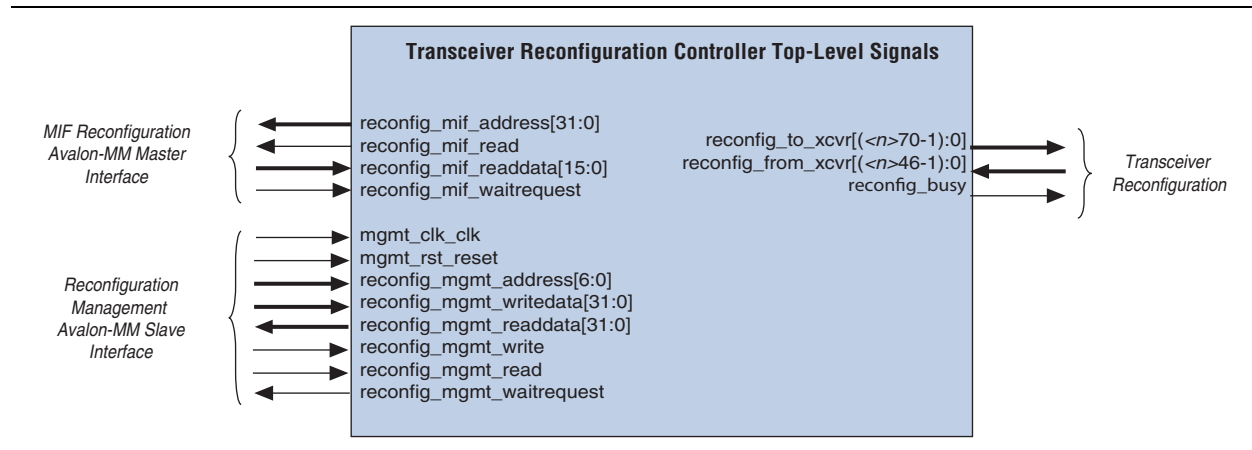
Table 10-4. General Options

Name	Value	Description
Device family	Arria V Stratix V	This IP core is available for Arria V and Stratix V devices. It is not functional for Arria V devices. It is available to establish the correct connections between the transceiver PHY and Transceiver Reconfiguration Controller.
Interface Bundles		
Number of reconfiguration interfaces	<IF>	Specifies the total number of reconfiguration interfaces that connect to the Transceiver Reconfiguration Controller. There is one interface for each channel and TX PLL. When you specify the parameters for a transceiver PHY, The message window displays the number of interfaces required.
Optional interface grouping	<Grp ₁ >, <Grp ₂ >, <Grp ₃ >	Specifies the grouping of reconfiguration interfaces as a comma-separated list with each integer indicating the total number of reconfiguration interfaces that are connected to a transceiver PHY instance. Leave this entry blank if all reconfiguration interfaces connect to the same transceiver PHY instance. Refer to “ Understanding Logical Channel Numbering ” on page 10-31 for more information about grouping interfaces.
Transceiver Calibration Functions		
Enable offset cancellation	On	When enabled, the Transceiver Reconfiguration Controller includes the offset cancellation functionality. This option is always on. Offset cancellation occurs automatically at power-up and runs only once.
Enable duty cycle calibration	On/Off	When enabled, this circuitry improves the duty cycle of the transceiver PHY IP core transmitters.
Enable auxiliary transmit (ATX) PLL calibration	On/Off	When enabled, an algorithm that improves the signal integrity of the ATX PLL is included in the Transceiver Reconfiguration Controller IP core.
Analog Features		
Enable Analog controls	On/Off	When enabled, RX and RX signal conditioning features are enabled.
Enable EyeQ block	On/Off	When enabled, you can use the EyeQ, the on-chip signal quality monitoring circuitry, to estimate the actual eye opening at the receiver.
Enable adaptive equalization (AEQ) block	On/Off	When enabled, the Transceiver Reconfiguration Controller includes logic to perform AEQ.
Reconfiguration Features		
Enable channel/PLL reconfiguration	On/Off	When enabled, the Transceiver Reconfiguration Controller includes logic to include both channel and PLL reconfiguration.
Enable PLL reconfiguration support block	On/Off	When enabled, the Transceiver Reconfiguration Controller includes logic to perform PLL reconfiguration.

Interfaces

This section describes interfaces for the Transceiver Reconfiguration Controller. [Figure 10-2](#) illustrates the top-level signals of the Transceiver Reconfiguration Controller.

Figure 10-2. Top-Level Signals of the Transceiver Reconfiguration Controller



 By default, the **Block Diagram** shown in the parameter editor labels the external pins with the *interface type* and places the *interface name* inside the box. The interface type and name are used in the Hardware Component Description File (`_hw.tcl`). If you click **Show signals**, the block diagram expands to show all of the signals of the component given the options currently selected in the parameter editor.

 For more information about `_hw.tcl` files refer to the [Component Interface Tcl Reference](#) in volume 1 of the *Quartus II Handbook*.

MIF Reconfiguration Management Avalon-MM Master Interface

[Table 10-5](#) describes the signals that comprise the dynamic reconfiguration interface. The Transceiver Reconfiguration Controller communicates with the PHY IP cores using this interface.

Table 10-5. MIF Reconfiguration Management Avalon-MM Master Interface

Signal Name	Direction	Description
<code>reconfig_mif_address[31:0]</code>	Output	This is the Avalon-MM address. This is a byte address.
<code>reconfig_mif_read</code>	Output	When asserted, signals an Avalon-MM read request.
<code>reconfig_mif_readdata[15:0]</code>	Input	The read data.
<code>reconfig_mif_waitrequest</code>	Input	When asserted, indicates that the MIF Avalon-MM slave is not ready to respond to a read request.

Transceiver Reconfiguration Interface

Table 10-6 describes the signals that comprise the dynamic reconfiguration interface. The Transceiver Reconfiguration Controller communicates with the PHY IP cores using this interface.

Table 10-6. Transceiver Reconfiguration Interface

Signal Name	Direction	Description
reconfig_to_xcvr[(<n>x70)-1:0] ⁽¹⁾	Output	Parallel reconfiguration bus from the Transceiver Reconfiguration Controller to the PHY IP core.
reconfig_from_xcvr[(<n>x46)-1:0] ⁽¹⁾	Input	Parallel reconfiguration bus from the PHY IP core to the Transceiver Reconfiguration Controller.
reconfig_busy	Output	When asserted, indicates that a reconfiguration operation is in progress and no further reconfiguration operations should be performed. You can monitor this signal to determine the status of the Transceiver Reconfiguration Controller. Alternatively, you can monitor the <code>busy</code> bit of the <code>control</code> and <code>status</code> registers of any reconfiguration feature to determine the status of the Transceiver Reconfiguration Controller.

Note to Table 10-6:

(1) <n> is the number of reconfiguration interfaces connected to the Transceiver Reconfiguration Controller.

Reconfiguration Interface Management Interface

The reconfiguration management interface is an Avalon-MM slave interface. You can use an embedded controller to drive this interface. Alternatively, you can use a finite state machine to control all Avalon-MM reads and writes to the Transceiver Reconfiguration Controller. This interface provides access to the Transceiver Reconfiguration Controller's Avalon-MM registers.



For more information about the Avalon-MM protocol, including timing diagrams, refer to the [Avalon Interface Specifications](#). Table 10-7 list the signals in reconfiguration management interface.

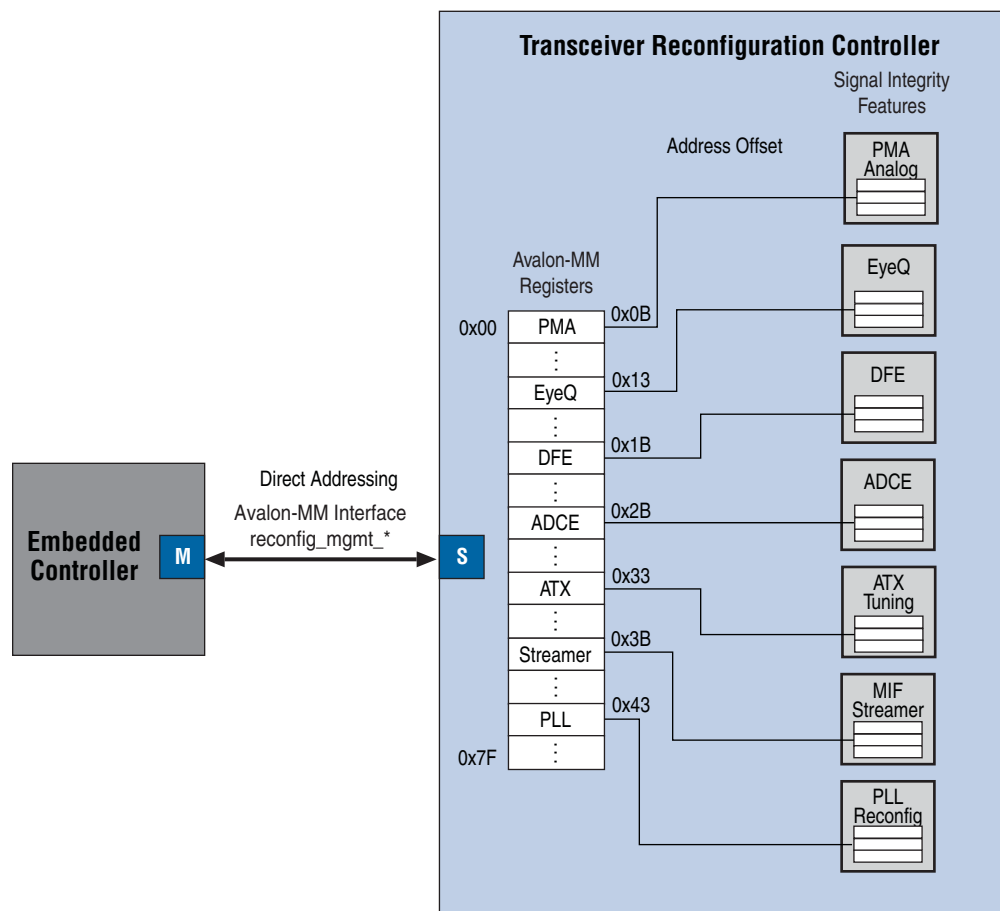
Table 10-7. Reconfiguration Management Interface

Signal Name	Direction	Description
mgmt_clk_clk	Input	Avalon-MM clock input. The frequency range for the <code>mgmt_clk_clk</code> is 100–125 MHz.
mgmt_rst_reset	Input	This signal resets the Transceiver Reconfiguration Controller. This signal is active high and level sensitive.
reconfig_mgmt_address[6:0]	Input	7-bit Avalon-MM address.
reconfig_mgmt_writedata[31:0]	Input	Input data.
reconfig_mgmt_readdata[31:0]	Output	Output data.
reconfig_mgmt_write	Input	Write signal. Active high.
reconfig_mgmt_read	Input	Read signal. Active high.

Reconfiguration Controller Memory Map

This section describes the memory map that control reconfiguration and signal integrity features. Each register-based feature has its own Avalon-MM address space within the Transceiver Reconfiguration Controller as [Figure 10-3](#) illustrates.

Figure 10-3. Memory Map of the Transceiver Reconfiguration Controller Registers



[Table 10-8](#) lists the address range for the Transceiver Reconfiguration Controller and the reconfiguration and signal integrity modules. It provides links to the sections describing the registers in each module.

Table 10-8. Transceiver Reconfiguration Controller Address Map (Part 1 of 2)

Address	Link
7'h08-7'h0C	“PMA Analog Controls” on page 10-11
7'h10-7'h14	“EyeQ” on page 10-12
7'h18-7'h1C	“DFE” on page 10-14
7'h28-7'h2C	“AEQ” on page 10-16
7'h30-7'h34	“ATX PLL Calibration” on page 10-17

Table 10-8. Transceiver Reconfiguration Controller Address Map (Part 2 of 2)

Address	Link
7'h38-7'h3C	“Streamer Module” on page 10-22
7'h40-7'h44	“PLL Reconfiguration” on page 10-18

Transceiver Calibration Functions

The Transceiver Reconfiguration Controller supports various calibration functions to enhance the performance and operation of any connected transceiver PHY IP core. This section describes the functionality of each calibration function. Refer to [Table 10-3 on page 10-5](#) for the resource utilization of these calibration functions.

Offset Cancellation

The offset cancellation function adjusts the offsets within the RX PMA and the CDR parameters for process variations to achieve optimal performance. Offset cancellation runs only once upon power-up. The RX buffers are unavailable while this function is running. This calibration feature is run automatically and enabled by default.

Duty Cycle Calibration

The duty cycle calibration function tunes the transmitter to minimize duty cycle distortion. Altera recommends that you enable this function for all transceiver PHY IP cores with a data ≥ 6 Gbps. Both the TX and RX buffers are unavailable while this function is running.



If you select a TX-only transceiver PHY, duty cycle calibration does not run. To run duty cycle calibration, you can instantiate an unused receiver channel.

Auxiliary Transmit (ATX) PLL Calibration

ATX calibration tunes the parameters of the ATX PLL for optimal performance. This function runs once after power up. You can rerun this function by writing into the appropriate memory-mapped registers.

You should enable ATX calibration for all transceiver PHY IP cores that use an ATX PLL. The RX buffer is unavailable while this function is running. You should run the ATX calibration after reconfiguring the PLL. You may need to rerun ATX calibration if you reset an ATX PLL and it does not lock after the specified lock time. When a design includes an ATX PLL, this calibration feature runs once upon powerup. After powerup, you must manually initiate the calibration function.

For more information on how to control the Auxiliary Transmit (ATX) PLL Calibration refer to [“ATX PLL Calibration” on page 10-17](#).

Refer to the [“Parameter Settings” on page 10-5](#) section for information on how to enabled these functions.

PMA Analog Controls

You can use the Transceiver Reconfiguration Controller to reconfigure the following analog controls:

- Differential output voltage (V_{OD})
- Pre-emphasis taps
- Receiver equalization control
- Receiver equalization DC gain

Table 10-9 lists the memory-mapped PMA analog registers that you can access using the reconfiguration management interface.



All undefined register bits are reserved.

Table 10-9. PMA Analog Registers

Recon-fig Addr	Bits	R/W	Register Name	Description
7'h08	[9:0]	RW	logical channel number	The logical channel number. Must be specified when performing dynamic updates. The Transceiver Reconfiguration Controller maps the logical address to the physical address.
7'h09	[9:0]	R	physical channel address	The physical channel address. The Transceiver Reconfiguration Controller maps the logical address to the physical address.
7'h0A	[9]	R	control and status	Error. When asserted, indicates an error. This bit is asserted if any of the following conditions occur: ■ The channel address is invalid. ■ The PHY address is invalid. ■ The PMA offset is invalid.
	[8]	R		Busy. When asserted, indicates that a reconfiguration operation is in progress.
	[1]	W		Read. Writing a 1 to this bit triggers a read operation.
	[0]	W		Write. Writing a 1 to this bit triggers a write operation.
7'h0B	[5:0]	RW	pma offset	Specifies the offset of the PMA analog setting to be reconfigured. Table 10-10 describes the valid offset values.
7'h0C	[6:0]	RW	data	Reconfiguration data for the PMA analog settings. Refer to Table 10-10 for valid data values.

 Refer to the *DC and Switching Characteristics for Stratix V Devices* in the *Stratix V Device Handbook* for the correspondence between the value specified by the reconfiguration data and the actual values that result from these assignments. The actual values are currently pending characterization of Stratix V silicon.

 All undefined register bits are reserved and must be set to 0.

Table 10-10. PMA Offsets and Values

Offset	Bits	R/W	Register Name	Description
0x0	[6:0]	RW	V ₀₀	V ₀₀ . The following encodings are defined: ■ 6'b000000:6'b111111:0–63
0x1	[5:0]	RW	Pre-emphasis pre-tap	The following encodings are defined: ■ 5'b00000–5'b10000: 0 ■ 5'b00001–5'b01111: -15 to -1 ■ 5'b10001–5'b11111: 1 to 15
0x2	[5:0]	RW	Pre-emphasis first post-tap	The following encodings are defined: ■ 5'b00000–5'b11111: 0–31
0x3	[5:0]	RW	Pre-emphasis second post-tap	The following encodings are defined: ■ 5'b00000–5'b10000: 0 ■ 5'b00001–5'b01111: -15 to -1 ■ 5'b10001–5'b11111: 1 to 15
0x10	[3:0]	RW	RX equalization DC gain	The following encodings are defined: ■ 3'b000–3'b111 :0–4
0x11	[3:0]	WO	RX equalization control	The following encodings are defined: ■ 4'b0000–4'b1111: 0–15
0x20	[0]	WO	Pre-CDR Reverse Serial Loopback	Writing a 1 to this bit enables S reverse serial loopback. Writing a 0 disables pre-CDR reverse serial loopback.
0x21	[0]	WO	Post-CDR Reverse Serial Loopback	Writing a 1 to this bit enables post-CDR reverse serial loopback. Writing a 0 disables post-CDR reverse serial loopback.

Refer to “Changing Transceiver Settings Using Register-Based Reconfiguration” on page 10-27 and “Changing Transceiver Settings Using Streamer-Based Reconfiguration” on page 10-28 for the procedures you can use to update PMA settings.

 Refer to *Application Note 645: Dynamic Reconfiguration of PMA Controls in Stratix V Devices* for an example demonstrating the use of the Transceiver Reconfiguration Controller.

EyeQ

EyeQ is a debug and diagnostic tool that analyzes the incoming data, including the receiver's gain, noise level, and jitter after the receive buffer.

EyeQ uses a phase interpolator and sampler to estimate the vertical and horizontal eye opening using the values that you specify for the horizontal phase and vertical height. (Refer to [Table 10-12 on page 10-14](#).) The phase interpolator generates a sampling clock and the sampler examines the data from the receiver output. The sampled data is deserialized and sent to the IP core where the PRBS checker determines the BER. As the phase interpolator output clock phase is shifted by small increments, the data error rate goes from high to low to high if the receiver is good. The number of steps of valid data is defined as the width of the eye. If none of the steps yields valid data, the width of the eye is equal to 0, which means the eye is closed.

[Table 10-11](#) lists the memory-mapped EyeQ registers that you can access using Avalon-MM reads and writes on reconfiguration management interface.



All undefined register bits are reserved.

Table 10-11. Eye Monitor Registers

Recon-fig Addr	Bits	R/W	Register Name	Description
7'h10	[9:0]	RW	logical channel number	The logical channel number. Must be specified when performing dynamic updates. The Transceiver Reconfiguration Controller maps the logical address to the physical address.
7'h12	[9]	R	control and status	Error. When asserted, indicates an invalid channel or address.
	[8]	R		Busy. When asserted, indicates that a reconfiguration operation is in progress.
	[1]	W		Read. Writing a 1 to this bit triggers a read operation.
	[0]	W		Write. Writing a 1 to this bit triggers a write operation.
7'h13	[5:0]	RW	eyeq offset	Specifies the 6-bit offset of the EyeQ register.
7'h14	[15:0]	RW	data	Reconfiguration data for the transceiver PHY registers.

Table 10-12 describes the EyeQ registers that you can access.



All undefined register bits are reserved and must be set to 0.

Table 10-12. EyeQ Offsets and Values

Offset	Bits	R/W	Register Name	Description
0x0	[1]	RW	Control	Writing a 1 to this bit enables the Eye monitor.
0x1	[5:0]	RW	Horizontal phase	Taken together, the horizontal phase and vertical height specify the Cartesian x-y coordinates of the point on the eye diagram that you want to sample. You can increment through 64 phases over 2 UI on the horizontal axis.
0x2	[5:0]	RW	Vertical height	Taken together, the horizontal phase and vertical height specify the Cartesian x-y coordinates of the point on the eye diagram that you want to sample. You can specify 64 heights over 2 UI on the vertical axis.

Refer to “[Changing Transceiver Settings Using Register-Based Reconfiguration](#)” on page 10-27 for the procedures you can use to control the Eye Monitor.

DFE

The DFE is an infinite impulse response filter (non-linear) that compensates for inter-symbol interference (ISI). Because the values of symbols previously detected are known, the DFE engine can estimate the ISI contributed by these symbols and cancel out this ISI by subtracting the predicted value from subsequent symbols. This mechanism allows DFE to boost the signal to noise ratio of the received data. You can use DFE in conjunction with the receiver's linear equalization and with the transmitter's pre-emphasis feature.

If you enable the DFE engine, you can determine the optimal settings by monitoring the BER of the received data at each setting and specify the DFE settings that yield the widest eye.

Table 10-13 lists the direct DFE registers that you can access using Avalon-MM reads and writes on reconfiguration management interface.



All undefined register bits are reserved.

Table 10-13. DFE Registers (Part 1 of 2)

Recon-fig Addr	Bits	R/W	Register Name	Description
7'h18	[9:0]	RW	logical channel address	The logical channel address. Must be specified when performing dynamic updates. The Transceiver Reconfiguration Controller maps the logical address to the physical address.

Table 10-13. DFE Registers (Part 2 of 2)

Recon-fig Addr	Bits	R/W	Register Name	Description
7'h1A	[9]	R	control and status	Error. When asserted, indicates an invalid channel or address.
	[8]	R		Busy. When asserted, indicates that a reconfiguration operation is in progress.
	[1]	W		Read. Writing a 1 to this bit triggers a read operation.
	[0]	W		Write. Writing a 1 to this bit triggers a write operation.
7'h1B	[5:0]	RW	dfe_offset	Specifies the 6-bit offset of the DFE register.
7'h1C	[15:0]	RW	data	Reconfiguration data for the transceiver PHY registers.

Table 10-14 describes the DFE registers that you can access to change DFE settings.



All undefined register bits are reserved and must be set to 0.

Table 10-14. DFE Offset and Values (Part 1 of 2)

Offset	Bits	R/W	Register Name	Description
0x0	[1]	RW	power on	A 0 to 1 transition on this bit triggers DFE calibration.
	[0]	RW	adaptation engine enable	Writing a 1 triggers the adaptive equalization engine.
0x1	[3:0]	RW	tap 1	Specifies the coefficient for the first post tap. The valid range is 0–15.
0x2	[3]	RW	tap 2 polarity	Specifies the polarity of the second post tap as follows: ■ 0: negative polarity ■ 1: positive polarity
	[2:0]	RW	tap 2	Specifies the coefficient for the second post tap. The valid range is 0–7.
0x3	[3]	RW	tap 3 polarity	Specifies the polarity of the third post tap as follows: ■ 0: negative polarity ■ 1: positive polarity
	[2:0]	RW	tap 3	Specifies the coefficient for the third post tap. The valid range is 0–7.
	[3]	RW	tap 4 polarity	Specifies the polarity of the fourth post tap as follows: ■ 0: negative polarity ■ 1: positive polarity
0x4	[2:0]	RW	tap 4	Specifies the coefficient for the fourth post tap.

Table 10-14. DFE Offset and Values (Part 2 of 2)

Offset	Bits	R/W	Register Name	Description
0x5	[3]	RW	tap 5 polarity	Specifies the polarity of the fifth post tap as follows: ■ 0: negative polarity ■ 1: positive polarity
	[2:0]	RW	tap 5	Specifies the coefficient for the fifth post tap. The valid range is 0–7.
0x6	[2:0]	RW	reference voltage level	Specifies the reference voltage: The following encodings are defined: ■ 3'b000: 0 mV ■ 3'b001: 35 mV ■ 3'b010: 55 mV ■ 3'b011: 70 mV ■ 3'b100: 110 mV ■ 3'b101: 150 mV ■ 3'b110: 200 mV ■ 3'b111: 1000 mV

AEQ

Adaptive equalization compensates for backplane losses and dispersion which degrade signal quality. You can choose to run the AEQ once at power up or to run it continuously to dynamically adapt to changing conditions. You can also use the AEQ to help control the four-stage continuous time linear equalizer (CTLE) which is a manual tool that compensates for backplane losses and dispersion.

Table 10-15 lists the direct AEQ registers that you can access using Avalon-MM reads and writes on reconfiguration management interface.



All undefined register bits are reserved.

Table 10-15. AEQ Registers (Part 1 of 2)

Recon-fig Addr	Bits	R/W	Register Name	Description
7'h28	[9:0]	RW	logical channel number	The logical channel number of the AEQ hardware to be accessed. Must be specified when performing dynamic updates. The Transceiver Reconfiguration Controller maps the logical address to the physical address.
7'h29	[9:0]	R	physical channel address	The physical channel address. The Transceiver Reconfiguration Controller maps the logical address to the physical address.

Table 10-15. AEQ Registers (Part 2 of 2)

Recon-fig Addr	Bits	R/W	Register Name	Description
7'h2A	[9]	R	control and status	Error. When asserted, indicates an error. This bit is asserted when the channel address is invalid.
	[8]	R		Busy. When asserted, indicates that a reconfiguration operation is in progress.
	[1]	W		Read. Writing a 1 to this bit triggers a read operation.
	[0]	W		Write. Writing a 1 to this bit triggers a write operation.
7'h2B	[3:0]	RW	aeq_offset	Specifies the address of the AEQ register to be read or written. Refer to Table 10-16 for details.
7'h2C	[15:0]	RW	data	Specifies the read or write data.

[Table 10-16](#) describes the AEQ registers that you can access to change AEQ settings.



All undefined register bits are reserved and must be set to 0.

Table 10-16. AEQ Offsets and Values

Offset	Bits	R/W	Register Name	Description	Default Value
0x0	[8]	R	adapt_done	When asserted, indicates that adaptation has completed.	1b'0
	[1:0]	RW	mode	Specifies the following address modes: ■ 2'b00: Low power manual equalization mode ■ 2'b01: One-time AEQ adaptation at power up ■ 2'b10: Perform continuous AEQ adaptation ■ 2'b11: Reserved	2'b00
0x2	[3:0]	R	equalization results	This is the value set by the automatic AEQ adaptation performed at startup. If you choose to perform manual equalization using the linear equalizer, you can use this value as a reference. Although automatic and manual equalization do not provide identical functionality, specifying this value enables manual equalization to approximate the original setting.	0x0000

Refer to [“Changing Transceiver Settings Using Register-Based Reconfiguration”](#) on [page 10-27](#) for the procedures you can use to control AEQ.

ATX PLL Calibration

This feature allows you to rerun ATX calibration after power up. The Transceiver Reconfiguration Controller automatically runs ATX calibration at power up.



You may need to rerun ATX calibration if you reset an ATX PLL and it does not lock after the specified lock time.

Table 10-17 lists the direct access ATX registers that you can access using Avalon-MM reads and writes on reconfiguration management interface.



All undefined register bits are reserved.

Table 10-17. ATX Tuning Registers

ATX Addr	Bits	R/W	Register Name	Description
7'h30	[9:0]	RW	logical channel number	The logical channel number. The Transceiver Reconfiguration Controller maps the logical address to the physical address.
7'h32	[9]	R	control and status	Error. When asserted, indicates an invalid channel or address. This bit is asserted after a write operation if the selected logical channel number selects a logical channel interface that is not connected to an ATX PLL. It is also be asserted if the tuning algorithm failed to converge on a working setting after a manual calibration.
	[8]	R		Busy. When asserted, indicates that a reconfiguration operation is in progress.
	[1]	W		Read. Writing a 1 to this bit triggers a read operation.
	[0]	W		Write. Writing a 1 to this bit triggers a write operation.
7'h33	[3:0]	RW	atx_offset	Specifies the 4-bit register address used for indirect accesses on the reconfiguration bus. Refer to Table 10-18 for offsets and values.
7'h34	[15:0]	RW	data	Reconfiguration data for the transceiver PHY registers.

Table 10-18 lists the ATX PLL tuning registers.

Table 10-18. ATX PLL Tuning Offsets and Values

Offset	Bits	R/W	Register Name	Description
0x0	[1]	RW	Control	Writing a 1 to this bit triggers ATX PLL calibration. This register self-clears. Unused bits of this register must be set to 0.

Refer to “Changing Transceiver Settings Using Register-Based Reconfiguration” on page 10-27 for the procedures you can use to control ATX tuning.

PLL Reconfiguration

You can use the PLL reconfiguration registers to change the reference clock input to the TX PLL or the clock data recovery (CDR) circuitry.

The PLL registers for dynamic reconfiguration feature are available when you select one of the following Stratix V transceiver PHY IP cores:

- Custom PHY IP Core
- Low Latency PHY IP Core

You can establish the number of possible PLL configurations on the **Reconfiguration** tab of the appropriate transceiver PHY IP core. The **Reconfiguration** tab allows you to specify up to five input reference clocks and up to four TX PLLs. You can also change the input clock source to the CDR PLL; up to five input clock sources are possible. If you plan to dynamically reconfigure the PLLs in your design, you must also enable **Allow PLL Reconfiguration** and specify the **Main TX PLL logical index** which is the PLL that the Quartus II software instantiates at power up. Figure 10-4 illustrates these parameters.



You must provide your own custom reset controller if you dynamically reconfigure the PLLs in your design.

For more information on the Stratix V reset sequence, refer to *Transceiver Reset Control in Stratix V Devices* in volume 3 of the *Stratix V Device Handbook*.

Figure 10-4. Reconfiguration Tab of Custom and Low Latency Transceiver PHYs

The screenshot shows the 'Reconfiguration' tab of the Quartus II software. It contains the following settings:

- PLL Reconfiguration:**
 - ☒ Allow PLL Reconfiguration
 - Number of TX PLLs: 2
 - Number of input clocks: 1
 - Main TX PLL logical index: 1
 - CDR PLL input clock source: 0
- TX PLL 0:**
 - PLL type: CMU
 - Base data rate: 3125 Mbps
 - Input clock frequency: 156.25 MHz
 - Selected input clock source: 0
- TX PLL 1:**
 - PLL type: CMU
 - Base data rate: 1250 Mbps
 - Input clock frequency: 62.5 MHz
 - Selected input clock source: 0
- Channel Interface:**
 - ☐ Enable Channel Interface

When you specify multiple PLLs, you must use the QSF assignment, `XCVR_TX_PLL_RECONFIG_GROUP`, to identify the PLLs within a reconfiguration group using the Assignment Editor. The `XCVR_TX_PLL_RECONFIG_GROUP` assignment identifies PLLs that the Quartus II Fitter can merge. You can assign TX PLLs from different transceiver PHY IP core instances to the same group.



You must create the `XCVR_TX_PLL_RECONFIG_GROUP` even if one transceiver PHY IP core instance instantiates multiple TX PLLs.

Table 10-19 lists the PLL reconfiguration registers that you can access using Avalon-MM read and write commands on reconfiguration management interface.



All undefined register bits are reserved.

Table 10-19. PLL Reconfiguration Registers

Addr	Bits	R/W	Register Name	Description
7'h40	[9:0]	RW	logical channel number	The logical channel number. Must be specified when performing dynamic updates. The Transceiver Reconfiguration Controller maps the logical address to the physical address.
7'h41	[9:0]	R	physical channel address	The physical channel address. The Transceiver Reconfiguration Controller maps the logical address to the physical address.
7'h42	[9]	R	control and status	When asserted, indicates an error. This bit is asserted if any of the following conditions occur: ■ The channel address is invalid. ■ The PHY address is invalid. ■ The address offset is invalid.
	[8]	R		MIF Busy. When asserted, indicates that a reconfiguration operation is in progress.
	[1]	W		Read. Writing a 1 to this bit specifies a read operation.
	[0]	W		Write. Writing a 1 to this bit specifies a write operation.
7'h43	[3:0]	RW	pll offset	Specifies the 4-bit register address used for indirect to the PLL registers on the reconfiguration bus. Refer to Table 10-20 for offsets and values.
7'h44	[15:0]	RW	data	Specifies the read or write data.

Table 10–20 lists the PLL reconfiguration registers.



All undefined register bits are reserved and must be set to 0.

Table 10–20. PLL Reconfiguration Offsets and Values

Offset	Bits	R/W	Name	Description
0x0	[2:0]	RW	logical refclk selection	When written initiates reference clock change to the logical reference clock indexed by bits [2:0]. For the Custom and Low Latency PHY IP cores, this index refers to the Number of input clocks on the Reconfiguration tab. You can specify up to 5 input clocks.
0x1	[2:0]	RW	logical PLL selection	When written initiates a clock generation block (CGB) switch to logical PLL indexed by bits [2:0]. For the Custom and Low Latency PHY IP cores, this index refers to the Number of TX PLLs selected on the Reconfiguration tab. You can specify up to 4 input clocks. If you set the Main TX PLL logical index to 0, the Quartus II software initializes your design using the first PLL defined.
0x2	[24:0]	RO	refclk physical mapping	Specifies the logical to physical refclk for current logical channel.
0x3	[14:0]	RO	PLL physical mapping	Specifies the logical to physical clock generation block word for current logical channel.

Channel and PLL Reconfiguration

You can use channel and PLL reconfiguration to dynamically reconfigure the channel and PLL settings in a transceiver PHY IP core. Among the settings that you can change dynamically are the data rate and interface width.

The Transceiver Reconfiguration Controller's Streamer Module implements channel and PLL reconfiguration. Refer to the [“Streamer Module” on page 10–22](#) for more information about this module.



Channel and PLL reconfiguration are available for the Custom and Low Latency PHY IP cores.

Channel Reconfiguration

If you turn on **Enable channel/PLL reconfiguration** in the Transceiver Reconfiguration Controller GUI, you can change the following channel settings:

- TX PMA settings
- RX PMA settings
- RX CDR input clock

- Reference clock inputs
- FPGA fabric transceiver width

When you select **Enable Channel Interface**, in the Custom and Low Latency Transceiver PHY GUIs, the default width of the FPGA fabric to transceiver interface increases for both the **Standard** and **10G** datapaths as follows:

- Standard datapath—The TX interface is 44 bits. The RX interface is 64 bits.
- 10G datapath—TX only, RX only, and duplex channels are all 64 bits.

However, depending upon the FPGA fabric transceiver width specified, only a subset of the 64 bits may carry valid data. Specifically, in the wider bus, only the lower $\langle n \rangle$ bits are used, where $\langle n \rangle$ is equal to the width of the FPGA fabric width specified in the transceiver PHY IP core. Table 10–21 illustrates this point for the 10G datapath, showing three examples where the FPGA fabric interface width is less than 64 bits.

Table 10–21. Channel Reconfiguration Bit Ordering

Number of Lanes	Specified FPGA Fabric Width (Total Bits)	Default Channel Width (Total Bits)	Used Bits
1	32 bits (32 bits)	64 bits/lane (64 bits)	Lane 0: [31:0]
2	40 bits (80 bits)	64 bits/lane (128 bits)	Lane 0: [39:0] Lane 1: [103:64]
3	40 bits (120 bits)	64 bits/lane (192 bits)	Lane 0: [39:0] Lane 1: [103:64] Lane 2: [167:128]

PLL Reconfiguration

If you turn on **Enable PLL reconfiguration support block** in the Transceiver Reconfiguration Controller GUI, you can change the following channel settings:

- TX PLL settings
- TX PLL selection



When you specify multiple PLLs, you must use the QSF assignment, `XCVR_TX_PLL_RECONFIG_GROUP`, to identify the PLLs within a reconfiguration group. The `XCVR_TX_PLL_RECONFIG_GROUP` assignment identifies PLLs that the Quartus II Fitter can merge.

Streamer Module

The Streamer module defines the following two modes for channel and PLL reconfiguration:

- Mode 0—MIF. Uses a memory initialization file (**.mif**) to reconfigure settings.
- Mode 1—Direct Write. Uses a series of Avalon-MM writes on the reconfiguration management interface to change settings. Table 10–9 lists the Streamer’s memory-mapped registers that you can access using Avalon-MM read and write commands on reconfiguration management interface.

 All undefined register bits are reserved.

Table 10–22. Streamer Module Registers

PHY Addr	Bits	R/W	Register Name	Description
7'h38	[9:0]	RW	logical channel number	The logical channel number. Must be specified when performing dynamic updates. The Transceiver Reconfiguration Controller maps the logical address to the physical address.
7'h39	[9:0]	R	physical channel address	The physical channel address. The Transceiver Reconfiguration Controller maps the logical address to the physical address.
7'h3A	[9]	R	control and status	Error. When asserted, indicates an error. This bit is asserted if any of the following conditions occur: ■ The channel address is invalid. ■ The PHY address is invalid. ■ The <code>offset</code> register address is invalid.
	[8]	R		Busy. When asserted, indicates that a reconfiguration operation is in progress.
	[3:2]	RW		Mode. The following encodings are defined: ■ 2'b00: MIF. This mode continuously reads and transfers a <code>.mif</code> file, which contains the reconfiguration data. ■ 2'b01: Direct Write. In this mode, you specify a logical channel, a register offset, and data. Depending on the logical channel specified, the Transceiver Reconfiguration Controller may mask some of the data specified to prevent read-only values that were optimized during startup, from being over-written. In particular, this mode protects the following settings: ■ Decision feedback equalization controls ■ RX buffer offset calibration adjustments ■ Duty cycle distortion adjustments ■ PMA clock settings ■ 2'b10: Reserved ■ 2'b11: Reserved
	[1]	W		Read. Writing a 1 to this bit triggers a read operation. This bit is self clearing.
	[0]	W		Write. Writing a 1 to this bit triggers a write operation. This bit is self clearing.
7'h3B	[15:0]	RW	offset	When the <code>MIF Mode = 2'b00</code> , the <code>offset</code> register specifies an internal MIF Streamer register. Refer to Table 10–23 for definitions of these registers. When <code>MIF Mode = 2'b01</code> , <code>offset</code> register specifies register in the transceiver
7'h3C	[31:0]	RW	data	When the <code>MIF Mode = 2'b00</code> , the <code>data</code> register stores read or write data for indirect access to the location specified in the <code>offset</code> register. When <code>MIF Mode = 2'b01</code> , <code>data</code> holds an update for transceiver to be dynamically reconfigured.

Table 10-23 lists the internal Streamer Module registers that you access to control and determine the status of a MIF operation.



All undefined register bits are reserved and must be set to 0.

Table 10-23. Streamer Module Internal MIF Register Offsets

Offset	Bits	R/W	Register Name	Description
0x0	[31:0]	RW	MIF base address	Specifies the MIF base address.
0x1	[2]	RW	Clear error status	Writing a 1 to this bit clears any error currently recorded in an indirect register. This register self clears. Any error detected in the error registers prevents MIF streaming. If an error occurs, you must clear the error register before restarting the Streamer.
	[0]	RW	Start MIF stream	Writing a 1 to this register, triggers a MIF streaming operation. This register self clears.
0x2	[4]	RO	MIF or Channel mismatch	When asserted, indicates the MIF type specified is incorrect. For example, the logical channel is duplex, but the MIF type specifies an RX only channel. The following 4 MIF types are defined: ■ Duplex ■ TX PLL (CMU) ■ RX only channel ■ TX only channel
	[2]	RO	PLL reconfiguration IP error	When asserted, indicates that an error occurred changing a refclk or clock generation block setting.
	[1]	RO	MIF opcode error	When asserted, indicates that an undefined opcode ID was specified in the .mif file, or the first entry in the .mif file was not a start of MIF opcode.
	[0]	RO	Invalid register access	When asserted, indicates that the offset register address specified is out of range.

The following sections describe operations in Streamer modes 0 and 1.

Mode 0 Streaming a MIF for Reconfiguration

In mode 0, you can stream the contents of a MIF containing the reconfiguration data to the transceiver PHY IP core instance. You specify this mode by writing a value of 2'b00 into bits 2 and 3 of the control and status register, as indicated in [Table 10-22 on page 10-23](#). Mode 0 simplifies the reconfiguration process because all reconfiguration data is stored in the MIF, which is streamed to the transceiver PHY IP in a single step.

The MIF can change PLL settings, reference clock inputs, or the TX PLL selection. After the MIF streaming update is complete, all transceiver PHY IP core settings reflect the value specified by the MIF. Refer to [“Streamer Based Reconfiguration” on page 10-28](#) for an example of a MIF update.

Mode 1 Avalon-MM Direct Writes for Reconfiguration

You specify this mode by writing a value of 2'b01 into bits 2 and 3 of the control and status register, as indicated in [Table 10-22 on page 10-23](#). In this mode, you can write directly to transceiver PHY IP core registers to perform reconfiguration. Refer to [“Direct Write Reconfiguration” on page 10-29](#) for an example of an update using mode 1. In mode 1, you can selectively reconfigure portions of the transceiver PHY IP core. Unlike mode 0, mode 1 allows you to write only the data required for a reconfiguration.

Stratix V MIF

The Stratix V MIF stores the reconfiguration data for the transceiver PHY IP cores. The Quartus II software automatically generates MIFs after each successful compilation. MIFs are stored in the **reconfig_mif** folder of the project's working directory. This folder stores all MIFs associated with the compiled project for each transceiver PHY IP core instance in the design. The parameter settings of PHY IP core instance reflect the currently specified MIF. You can store the MIF in an on-chip ROM or any other type of memory. This memory must connect to the MIF reconfiguration management interface.

[Example 10-1](#) shows file names for the .mif files for a design with two channels. This design example includes two transceiver PHY IP core instances running at different data rates. Both transceiver PHY IP core instances have two TX PLLs specified to support both 1 Gbps and 2.5 Gbps data rates. The Quartus II software generates two TX PLL .mif files for each PLL. The difference between the .mif files is the PLL reference clock specified. To dynamically reconfigure the channel from the initially specified data rate to a new data rate, you can use the MIF streaming function to load the other .mif.



When reconfiguration is limited to a few settings, you can create a partial .mif that only includes the settings that must be updated.

Example 10-1. Quartus II Generated MIF Files

```
<project_dir>/reconfig_mif/inst0_1g_channel.mif  
<project_dir>/reconfig_mif/inst0_1g_txpll0.mif  
<project_dir>/reconfig_mif/inst0_1g_txpll1.mif  
<project_dir>/reconfig_mif/inst0_2p5g_channel.mif  
<project_dir>/reconfig_mif/inst0_2p5g_txpll0.mif  
<project_dir>/reconfig_mif/inst0_2p5g_txpll1.mif
```

MIF Format

The Stratix-V MIF file is organized into records where each record contains the information necessary to carry out the reconfiguration process. There are two types of records: non-data records and data records. A MIF can contain a variable number of records, depending on the target transceiver channel. Both data records and non-data records are 16-bits long.

For both record types the high-order 5 bits represent the length field. A length field of 5'b0, indicates a non-data record which contains an opcode. A length field that is not zero indicates a data record.

For a non-data record, the opcode is represented by the lower 5-bits in the record. Table 10-24 lists the supported opcodes and describes the data content.

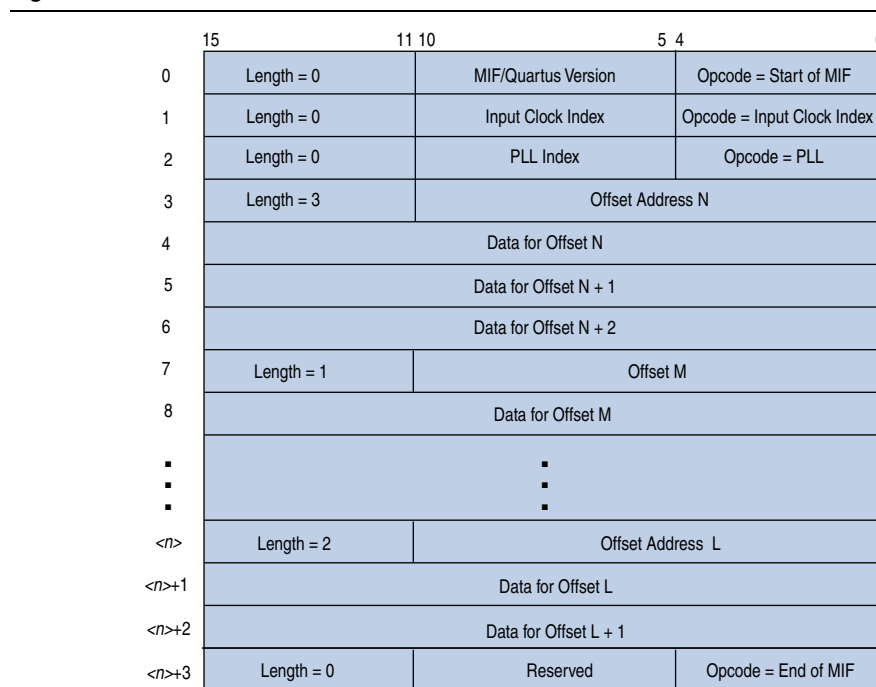
Table 10-24. Opcodes for MIF Files

Opcode	Opcode Description
5'b00000	Reserved
5'b00001	Start of MIF
5'b00010	Channel format indicator specifying the MIF channel type. The following encodings are defined: 2'b00: Duplex channel 2'b01: TX PLL (CMU) 2'b10: RX only channel 2'b11: TX only channel
5'b00011	CDR Input Clock switch
5'b00100	PLL switch
5'b00101-5'b11110	Reserved
5'b11111	End of MIF (EOM)

For data records, the low-order 11 bits provide a logical offset address. In this case, the length field indicates the number of data records that are written into the specified address. For example, if the length field is set to two, the next two records belong to the data record and are written into the offset address.

Figure 10-5 provides an example of a typical Stratix V MIF format. Entries 3, 7, and <n> are data records.

Figure 10-5. MIF File Format



Procedures for Reconfiguration

As Table 10-1 indicates, some features can only be reconfigured using register-based accesses, some features can only be reconfigured using MIF-based accesses, and some features can be reconfigured using either access mode. The following sections discuss both modes.

Changing Transceiver Settings Using Register-Based Reconfiguration

In register-based mode, you use a sequence of Avalon-MM writes and reads to update individual transceiver settings. The following section describes how to perform a register-based reconfiguration read and write.

Register-Based Write

Complete the following steps to perform a register-based write:

1. Read the control and status register busy bit (bit 8) until it is clear.
2. Write the logical channel number of the channel to be updated to the logical channel number register.
3. Write the *<feature>* offset address.
4. Write the appropriate data value to the data register.
5. Write the control and status register write bit to 1'b1.
6. Read the control and status register busy bit. Continue to read the busy bit while its value is one.
7. When busy = 0, the Transceiver Reconfiguration Controller has updated the logical channel specified in Step 2 with the data specified in Step 3.

Example 10-2 shows a reconfiguration that changes the logical channel o V_{OD} setting to 40.

Example 10-2. Register-Based Write of Logical Channel 0 V_{OD} Setting

```
#Setting logical channel 0
write_32 0x8 0x0

#Setting offset to VOD
write_32 0xB 0x0

#Setting data register to 40
write_32 0xC 0x28

#Writing all data
write_32 0xA 0x1
```

Register-Based Read

Complete the following steps for a read:

1. Read the control and status register busy bit (bit 8) until it is clear.
2. Write the logical channel number of the channel to be read to the logical channel number register.
3. Write the *<feature>* offset address.

4. Write the control and status register read bit to 1'b1.
5. Read the control and status register busy bit. Continue to read the busy until the value is zero.
6. Read the data register to get the data.

Example 10-3 illustrates a read of the pre-emphasis pretap value for logical channel 2.

Example 10-3. Register-Based Read of Logical Channel 2 Pre-Emphasis Pretap Setting

```
#Setting logical channel 2
write_32 0x8 0x2

#Setting offset to pre-emphasis pretap
write_32 0xB 0x1

#Writing the logical channel and offset for pre-emphasis pretap
write_32 0xA 0x1

#Reading data register for the pre-emphasis pretap value
read_32 0xC
```

Changing Transceiver Settings Using Streamer-Based Reconfiguration

The Streamer's registers allow you to change to the PCS datapath settings, clock settings, and PLL parameters by reading the new settings from an on- or off-chip ROM. "[Streamer Module Registers](#)" on [page 10-23](#) lists the Streamer's memory-mapped registers that you can access using Avalon-MM read and write commands on reconfiguration management interface.

The following sections show how to change transceiver settings using Streamer modes 0 and 1.

Streamer Based Reconfiguration

Follow these steps to reconfigure a transceiver setting by streaming the contents of a MIF file through the Streamer Module.

1. Write the logical channel number to the Streamer logical channel register.
2. Write MIF mode, 2'b00, to the Streamer control and status register **mode** bits.
3. Write the MIF base address, 0x0, to the Streamer offset register.
4. Write the base address of the MIF file to the Streamer data register.
5. Write the Streamer control and status register write bit to 1'b1 to initiate a write of all the data set in in the previous steps.
6. Write to the Streamer internal offset register with the value to start a MIF stream, 0x1.
7. Write the Streamer internal data register with the value 0x1 to setup the streaming of the MIF.
8. Write to the Streamer control and status register to 1'b1, to initiate the streaming operation.
9. Read the control and status register busy bit. When the busy bit is deasserted, the MIF streaming operation has completed.

Example 10-4 illustrates the reconfiguration of logical channel 0 using a MIF with a base address of 0x100.

Example 10-4. Reconfiguration of Logical Channel 0 Using A MIF

```
#Setting logical channel 0
write_32 0x38 0x0
#Setting Streamer mode to 0
write_32 0x3A 0x0
#Setting Streamer offset register to the MIF base address (0x0)
write_32 0x3B 0x0
#Setting data register with the MIF base address
write_32 0x3C 0x100
#Writing all data to the Streamer
write_32 0x3A 0x1
#Setting Streamer Module offset for Start MIF stream
write_32 0x3B 0x1
#Setting data register with 0x1 to setup for streaming
write_32 0x3C 0x1
#Writing all data to the Streamer to start streaming the MIF
write_32 0x3A 0x1
#Read the busy bit to determine when the write has completed
read_32 0x3A
```

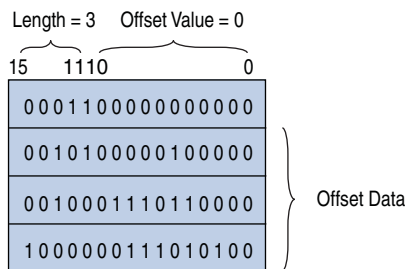
Direct Write Reconfiguration

Follow these steps to reconfigure a transceiver setting using a series of Avalon-MM direct writes.

1. Write the logical channel number to the Streamer logical channel register.
2. Write Direct Mode, 2'b01, to the Streamer control and status register mode bits.
3. Write the offset address to the Streamer offset register.
4. Write the offset data to the Streamer data register.
5. Write the Streamer control and status register write bit to 1'b1 to initiate a write of all the data set in the previous steps.
6. Repeat steps 3 through 5 if the offset data length is greater than 1. Increment the offset value by 1 for each additional data record.
7. Read the control and status register busy bit. When the busy bit is deasserted, the operation has completed.

In Steps 3 and 4, you must specify an offset value and offset data. You can determine the values of the offset address and offset data by examining the data records specified in either the channel or PLL MIFs. Figure 10-6 shows a sample MIF.

Figure 10-6. Sample MIF



For the sample data record in Figure 10-6, the length field specifies three data records. The offset value is 0, as indicated by bits 10-0. The offset data are the three subsequent entries. Example 10-5 performs a direct write in Streamer mode 1. This example writes the sample MIF in Figure 10-6 into the Streamer module which writes this data to logical channel 0.

Example 10-5. Streamer Mode 1 Reconfiguration

```
#Setting logical channel 0
write_32 0x38 0x0

#Setting Streamer to mode to 1
write_32 0x3A 4'b0100

#Setting Streamer offset register to the offset address
#In the example record, the first offset address is 0x0
write_32 0x3B 0x0

#Setting data register with the first data record
write_32 0x3C 16'b0010100000100000

#Writing first data to the Streamer
write_32 0x3A 0x1

#Incrementing Streamer offset register offset address
write_32 0x3B 0x1

#Setting data register with the second data record
write_32 0x3C 16'b0010001110110000

#Writing second data to the Streamer
write_32 0x3A 0x1

#Incrementing Streamer offset register offset address
write_32 0x3B 0x2
```

Example 10-5. (continued)

```
#Setting data register with the third data record
write_32 0x3C 16'b1000000111010100

#Writing third data record to the Streamer
write_32 0x3A 0x1

#Read the busy bit to determine when the operation completes
read_32 0x3a
```

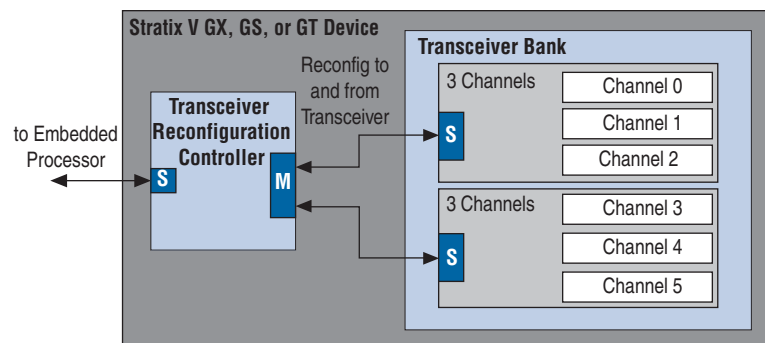
Understanding Logical Channel Numbering

This discussion of channel numbering, uses the following definitions:

- Reconfiguration interface—A bundle of signals that connect the Transceiver Reconfiguration Controller to a transceiver PHY data channel or TX PLL.
- Logical channels—An abstract representation of a channel or TX PLL that does not include physical location information.
- Bonded channel—A channel that shares a clock source with at least one other channel.
- Physical channel—The physical channel associated with a logical channel.

Figure 10-7 illustrates the connections between the Transceiver Reconfiguration Controller and a transceiver bank after running the Quartus II Fitter.

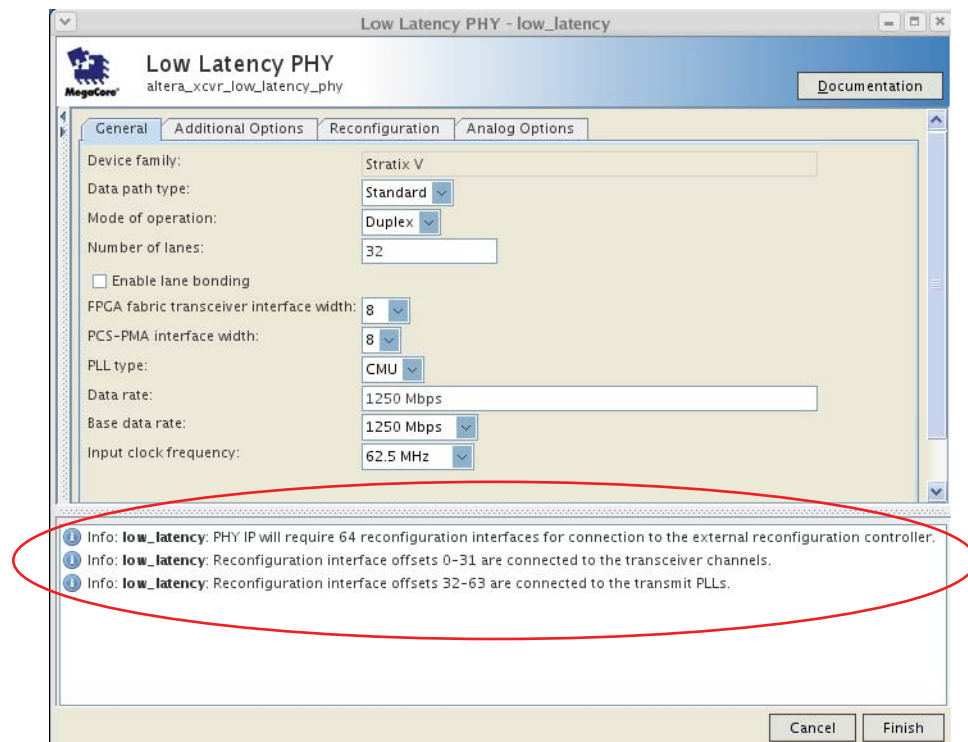
Figure 10-7. Post-Fit Connectivity



The transceiver PHY IP cores create a separate reconfiguration interface for each channel and each TX PLL. Each transceiver PHY IP core reports the number of reconfiguration interfaces it requires in the message pane of its GUI. You must take note of this number so that you can enter it as a parameter in the Transceiver Reconfiguration Controller.

Figure 10-8 shows the Low Latency PHY IP core GUI specifying 32 channels. The message pane indicates that reconfiguration interfaces 0-31 are for the transceiver channels and reconfiguration interfaces 32-63 are for the TX PLLs.

Figure 10-8. Low Latency Transceiver PHY Example



 After Quartus II compilation, many of the interfaces are merged.

Figure 10-9 illustrates the GUI for the Transceiver Reconfiguration Controller. To connect the Low Latency PHY IP core instance to the the Transceiver Reconfiguration Controller, you would enter 64 for **Number of reconfiguration interfaces**. You would not need to enter any values for the **Optional interface grouping** parameter because all of the interfaces belong to the same transceiver PHY IP core instance.

Figure 10-9. Transceiver Reconfiguration Controller Interface Bundles

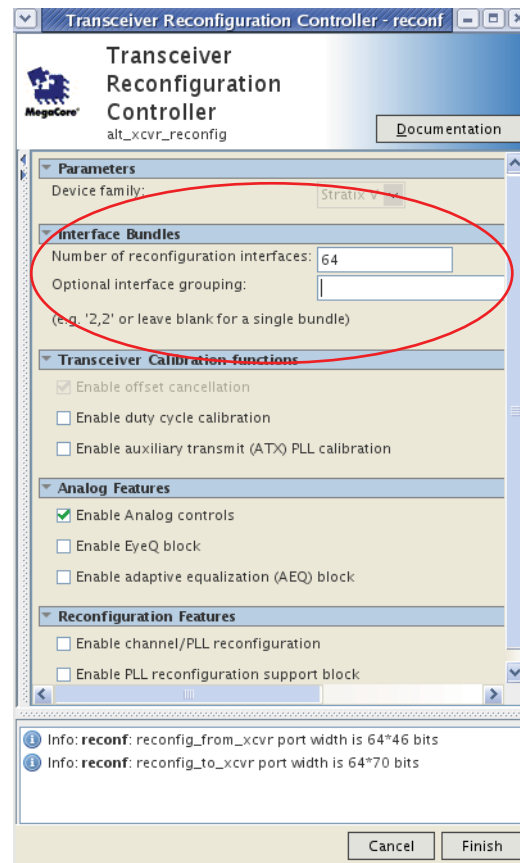


Figure 10-10 on page 10-34 shows a design with two transceiver PHY IP core instances, each with four channels. For this design you would enter 16 for the **Number of reconfiguration interfaces** and 8, 8 for the **Optional interface grouping** parameter.

Depending upon the transceiver PHY IP core and the parameters specified, the number of reconfiguration interfaces varies. For a single-channel, RX-only transceiver instance, there is a single reconfiguration interface. For example, one reconfiguration interface is created for a single-channel Low Latency PHY setup as a RX only channel. Two reconfiguration interfaces are created for a single-channel Custom PHY setup as a duplex channel. The reconfiguration interfaces do not appear as separate buses, but as a single bus of concatenated reconfiguration interfaces, that grows linearly with the number of reconfiguration interfaces.

Although you must create a separate logical reconfiguration interface for each PHY IP core instance, when the Quartus II software compiles your design, it reduces original number of logical interfaces by merging them. Allowing the Quartus II software to merge reconfiguration interfaces gives the Fitter more flexibility in placing transceiver channels. However, the logical channel number remains the same.



You cannot use SignalTap™ to observe the reconfiguration interfaces.

You do not have to assign numbers to the reconfiguration interfaces. The logical interface numbering is determined by the order of the interfaces in the connection between the transceiver PHY IP and the Transceiver Reconfiguration Controller.

Two PHY IP Core Instances Each with Four Bonded Channels

When two transceiver PHY instances, each with four bonded channels, are connected to a Transceiver Reconfiguration Controller, the reconfiguration buses of the two instances are concatenated. Figure 10-10 and Table 10-25 show the order and numbering of reconfiguration interfaces. The Quartus II software assigns the data channels logical channel numbers 0 to 3 for each transceiver PHY instance. The Quartus II software assigns the TX PLLs logical channel numbers 4 to 7 for each transceiver PHY instance. During Quartus II place and route, the Fitter maps the four logical TX PLLs in each transceiver PHY instance to a single physical TX PLL.

Figure 10-10. Interface Ordering with Multiple Transceiver PHY Instances

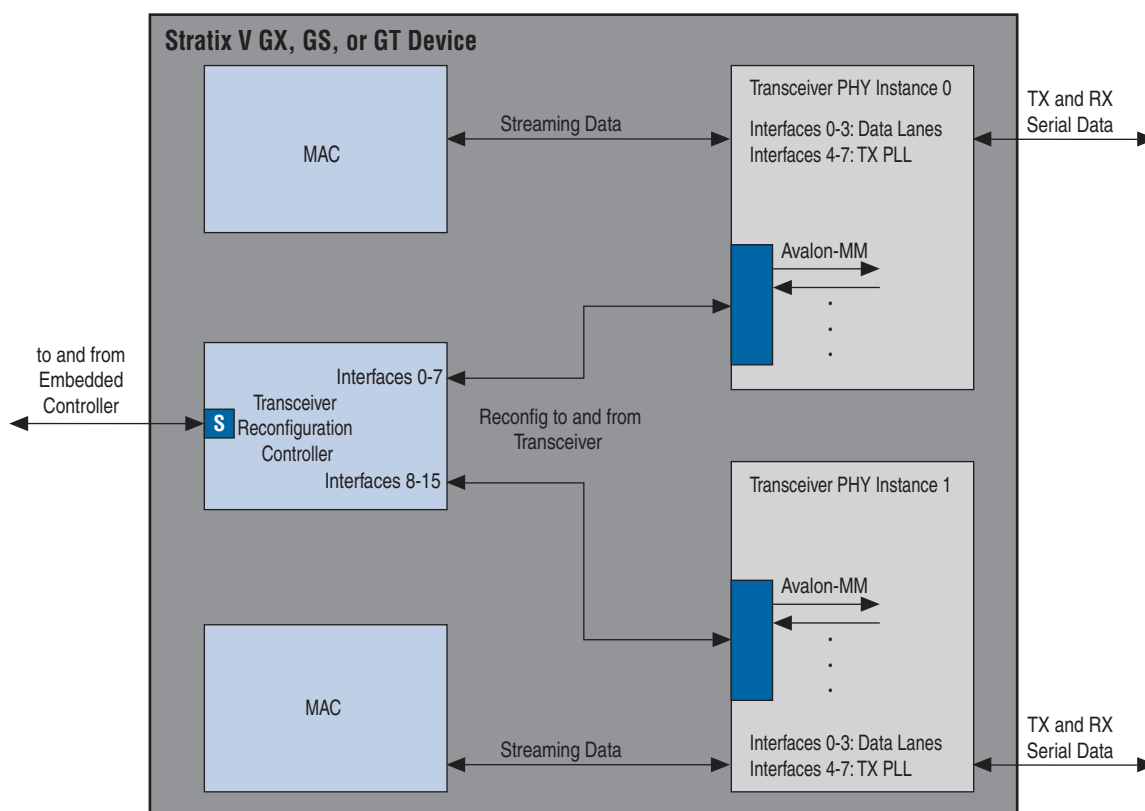


Table 10–25. Channel Ordering for Concatenated Transceiver Instances

Logical Interface Number	PHY Instance, Interface, or PLL
0–3	Instance 0, interfaces 0–3.
4–7	Instance 0, TX PLL. The Fitter assigns all 4 logical TX PLLs to a single physical PLL.
8–11	Instance 1, interfaces 0–3.
12–15	Instance 1, TX PLL. The Fitter assigns all 4 logical TX PLLs to a single physical PLL.

One PHY IP Core Instance with Eight Bonded Channels

This example requires the Quartus II Fitter to place channels in two, contiguous transceiver banks. To preserve flexibility for the Fitter, each channel and TX PLL is numbered separately. During place and route, the Fitter maps the eight logical TX PLLs to a single physical TX PLL.

Table 10–26 illustrates the logical channel numbering. In this table, logical address 0 accesses data channel 0 and logical address 8 accesses the TX PLL for data channel 0; logical address 1 accesses data channel 1 and logical address 9 accesses the TX PLL for data channel 1, and so on. In simulation, to reconfigure the TX PLL for channel 0, specify logical address 8 in the Streamer module’s logical channel number. The Streamer module maps the logical channel to the physical channel which would be the same value for all eight channels.

Table 10–26. Initial Number of Eight Bonded Channels

Channel	Logical Channel Number
Channel 0	0
Channel 1	1
Channel 2	2
Channel 3	3
Channel 4	4
Channel 5	5
Channel 6	6
Channel 7	7
CMU 0	8
CMU 1	9
CMU 2	10
CMU 3	11
CMU 4	12
CMU 5	13
CMU 6	14
CMU 7	15



Because all of the channels in a transceiver bank share a PLL, this original numbering allows the Fitter to select the optimal CMU PLL from a placement perspective by considering all of the TX PLLs in the bank.

Table 10-27 shows the channel numbers for post-Fitter and hardware simulations. At this point, you should have assigned channels to pins of the Stratix V device.

Table 10-27. Post-Fit Logical Channel Numbers for Eight Bonded Channels

Channel	Logical Channel Number
Channel 0	0
Channel 1	1
Channel 2	2
Channel 3	3
CMU (0-4)	8-12
Channel 4	4
Channel 5	5
CMU (5-7)	13-15
Channel 6	6
Channel 7	7

Two PHY IP Core Instances Each with Non-Bonded Channels

Non-bonded channels do not share TX PLLs. For each transceiver PHY IP core instance, the Quartus II software assigns the data channels sequentially beginning at logical address 0 and assigns the TX PLLs the subsequent logical addresses.

Table 10-28 illustrates the logical channel numbering for two transceiver PHY IP cores, one with 4 channels and one with 2 channels.

Table 10-28. Initial Number of Eight Bonded Channels

Instance	Channel	Logical Channel Number
Instance 0	Channel 0	0
	Channel 1	1
	Channel 2	2
	Channel 3	3
	CMU 0	4
	CMU 1	5
	CMU 2	6
	CMU 3	7
Instance 1	Channel 0	8
	Channel 1	9
	CMU 0	10
	CMU 1	11

Reconfiguration Controller to PHY IP Connectivity

You can connect a single Transceiver Reconfiguration Controller to all of the transceiver channels and PLLs in your design. You can also use multiple Transceiver Reconfiguration Controllers to facilitate placement and routing of the FPGA. However, the three, upper or lower contiguous channels in a transceiver bank must be connected to the same reconfiguration controller.

Figure 10-11 illustrates connections between the Transceiver Reconfiguration Controller and transceiver channels after Quartus II compilation.

Figure 10-11. Correct Connections

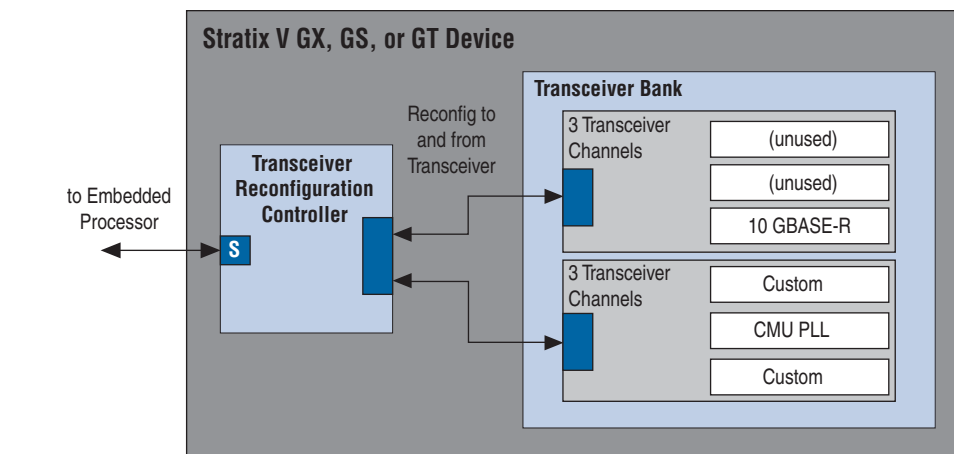
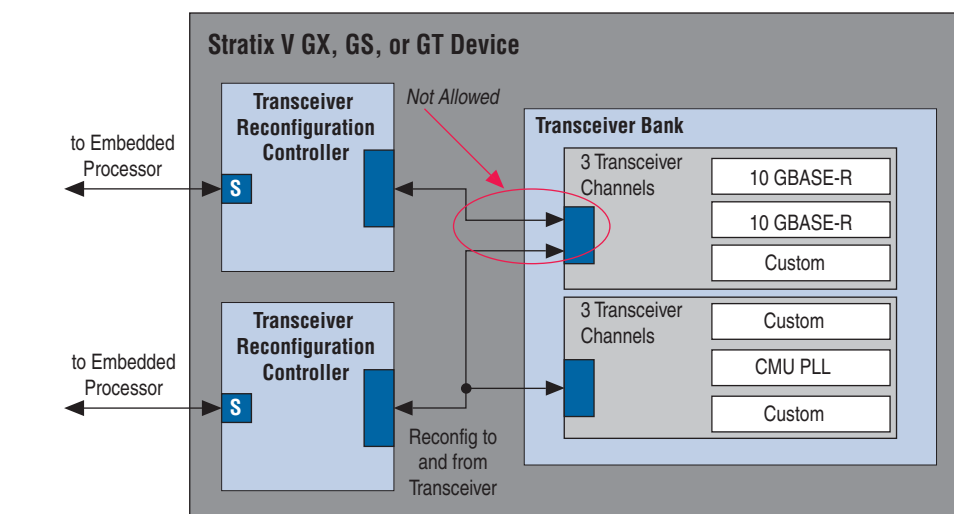


Figure 10-12 illustrates incorrect connections between two Transceiver Reconfiguration Controllers and six transceiver channels. Two Transceiver Reconfiguration Controllers cannot access a single reconfiguration interface because there is no arbitration logic to prevent concurrent access. The configuration shown in Figure 10-12 results in a Quartus II compilation error.

Figure 10-12. Incorrect Connections



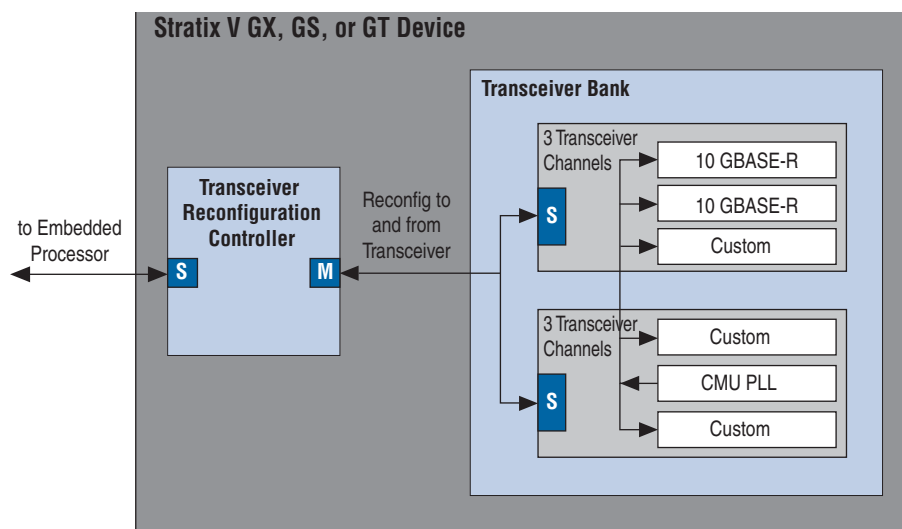
Merging TX PLLs In Multiple Transceiver PHY Instances

The Quartus II Fitter can merge the TX PLLs for multiple transceiver PHY IP cores under the following conditions:

- The PLLs connect to the same reset pin.
- The PLLs connect to the same reference clock.
- The PLLs connect to the same Transceiver Reconfiguration Controller .

Figure 10-13 illustrates a design where the CMU PLL in channel 1 provides the clock to three Custom PHY channels and two 10GBASE-R PHY channels.

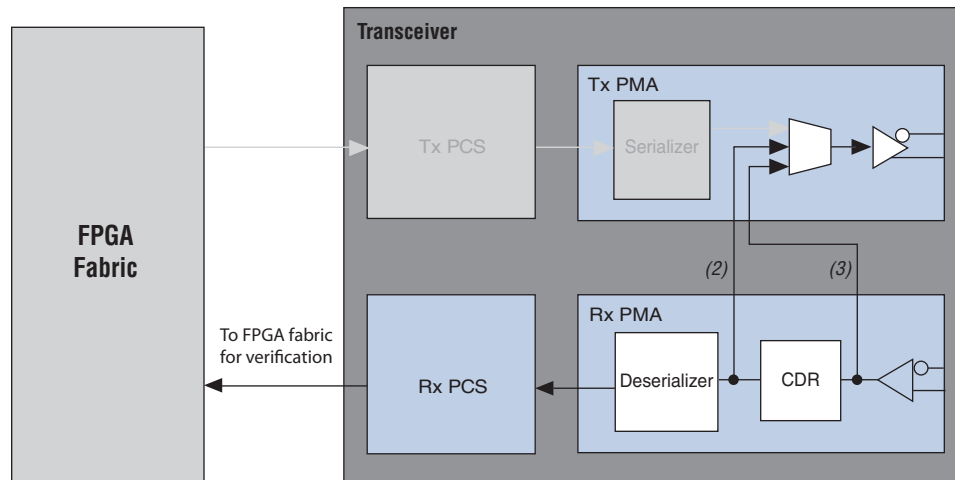
Figure 10-13. PLL Shared by Multiple Transceiver PHY IP Cores in a Single Transceiver Bank



Loopback Modes

You can enable the pre- and post-CDR reverse serial loopback modes by writing the appropriate bits of the Transceiver Reconfiguration Controller `pma_offset` register described in [Table 10-9 on page 10-11](#). In pre-CDR mode, data received through the RX input buffer is looped back to the TX output buffer. In post-CDR mode, received data passes through the RX CDR and then looped back to the TX output buffer. The RX data is also available to the FPGA logic. In the TX channel, only the TX buffer is active. [Figure 10-14](#) illustrates these modes.

Figure 10-14. Pre- and Post-CDR Reverse Serial Loopback Paths (1)



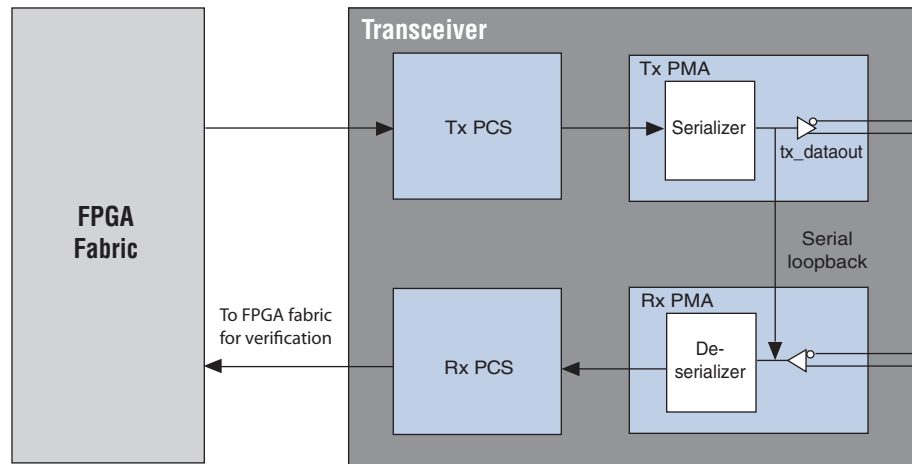
Notes to Figure 10-14:

- (1) Grayed-out blocks are not active in these modes.
- (2) Post-CDR reverse serial loopback path.
- (3) Pre-CDR reverse serial loopback path.

In addition to the pre-CDR and post-CDR loopback modes available in the Transceiver Reconfiguration Controller register map, all the of PHYs, with the exception of PCI Express, support serial loopback mode in both Stratix IV and Stratix V devices. This mode is enabled by writing the `phy_serial_loopback` register (0x061) using the Avalon-MM PHY management interface. PCI Express supports reverse parallel loopback mode as required by the *PCI Express Base Specification*.

Figure 10-15 shows the datapath for serial loopback. The data from the FPGA fabric passes through the TX channel and is looped back to the RX channel, bypassing the RX buffer. The received data is available to the FPGA logic for verification. Using the serial loopback option, you can check the operation of all enabled PCS and PMA functional blocks in the TX and RX channels. When serial loopback is enabled, the TX channel sends the data to both the tx_serial_data output port and the RX channel.

Figure 10-15. Serial Loopback



Previously, Altera provided the ALTGX megafunction as a general purpose transceiver PHY solution. The current release of the Quartus II software includes protocol-specific PHY IP cores that simplify the parameterization process.

The design of these protocol-specific transceiver PHYs is modular and uses standard interfaces. An Avalon-MM interface provides access to control and status registers that record the status of the PCS and PMA modules. Consequently, you no longer must include signals in the top level of your transceiver PHY to determine the status of the serial RX and TX interfaces. Using standard interfaces to access this device-dependent information should ease future migrations to other device families and reduce the overall design complexity. However, to facilitate debugging, you may still choose to include some device-dependent signals in the top level of your design during the initial simulations or even permanently. All protocol-specific PHY IP in Stratix V devices also include embedded controls for post-reset initialization which are available through the Avalon-MM interface.

For Stratix IV devices, the location of the transceiver dynamic reconfiguration logic is design dependent. In general, reconfiguration logic is integrated with the transceiver channels for simple configurations and is separately instantiated for more complex designs that use a large number of channels or instantiate more than one protocol in a single transceiver quad. For Stratix V devices, transceiver dynamic reconfiguration is always performed using the separately instantiated Transceiver Reconfiguration Controller.

Control of loopback modes is also different in Stratix IV and Stratix V devices. For Stratix IV devices, you must select loopback options in the using the parameter editor. For Stratix V devices, you control loopback modes through Avalon-MM registers.

Table 11–1 outlines these differences.

Table 11–1. Controlling Loopback Modes in Stratix IV and Stratix V Devices

Loopback Mode	Stratix IV	Stratix V
Serial loopback	On the Loopback tab of the ALTGX parameter editor, instantiate the <code>rx_serialpbken</code> signal by selecting the Serial loopback option. Drive this signal to 1 to put the transceiver in serial loopback mode.	Use the Avalon-MM PHY management interface to set the appropriate bit in the <code>phy_serial_loopback</code> register (0x061).
Reverse serial loopback (pre- and post-CDR)	On the Loopback tab of the ALTGX parameter editor, select either pre-CDR or post-CDR loopback and regenerate the ALTGX IP core.	Update the appropriate bits of the Transceiver Reconfiguration Controller <code>tx_rx_word_offset</code> register to enable the pre- or post-CDR reverse serial loopback mode. Refer to Table 10–9 on page 10–11 for more information.

This chapter enumerates the differences between the ALTGX megafunction for use with Stratix IV GX devices and the protocol-specific transceiver PHYs for use with Stratix V GX devices in the current release. It includes the following topics:

- [Dynamic Reconfiguration of Transceivers](#)
- [XAUI PHY](#)
- [PHY IP Core for PCI Express PHY \(PIPE\)](#)
- [Custom PHY](#)

Dynamic Reconfiguration of Transceivers

This section covers dynamic reconfiguration of transceivers in Stratix V and Stratix IV devices. Dynamic reconfiguration interface is completely new in Stratix V devices. You cannot automatically migrate a dynamic reconfiguration solution from Stratix IV to Stratix V devices.

Dynamic Reconfiguration for Stratix V Transceivers

Stratix V devices that include transceivers must use the Altera Transceiver Reconfiguration Controller that contains the offset cancellation logic to compensate for variations due to PVT. Initially, each transceiver channel and each TX PLL has its own parallel, dynamic reconfiguration bus, named `reconfig_from_xcvr[45:0]` and `reconfig_to_xcvr[69:0]`. The reconfiguration bus includes Avalon-MM signals to read and write registers and memory and test bus signals. When you instantiate a transceiver PHY in a Stratix V device, the transceiver PHY IP core provides informational messages specifying the number of required reconfiguration interfaces in the message pane as [Example 11-1](#) illustrates.

Example 11-1. Informational Messages for the Transceiver Reconfiguration Interface

```
PHY IP will require 5 reconfiguration interfaces for connection to the external
reconfiguration controller.
Reconfiguration interface offsets 0-3 are connected to the transceiver channels.
Reconfiguration interface offset 4 is connected to the transmit PLL.
```

Although you must initially create a separate reconfiguration interface for each channel and TX PLL in your design, when the Quartus II software compiles your design, it reduces the number of reconfiguration interfaces by merging reconfigurations interfaces. The synthesized design typically includes a reconfiguration interface for three channels. Allowing the Quartus II software to merge reconfiguration interfaces gives the Fitter more flexibility in placing transceiver channels.

Dynamic Reconfiguration for Stratix IV Transceivers

Stratix IV devices that include transceivers must use the ALTGX_RECONFIG IP core to implement dynamic reconfiguration. The ALTGX_RECONFIG IP core always includes the following two serial buses:

- `reconfig_from[<n>16:0]` — this bus connects to all the channels in a single quad. `<n>` is the number of quads connected to the ALTGX_RECONFIG IP core.

- `reconfig_togxb[3:0]`—this single bus connects to all transceiver channels.

If you select additional functionality in the parameter editor for the ALTGX_RECONFIG IP core, the IP core adds signals to support that functionality. For more information about the ALTGX_RECONFIG IP core, refer to *ALTGX_RECONFIG Megafunction User Guide for Stratix IV Devices* in volume 3 of the *Stratix IV Device Handbook*.

XAUI PHY

This section lists the differences between the parameters and signals for the XAUI PHY IP core and the ALTGX megafunction when configured in the XAUI functional mode.

Parameter Differences

Table 11-2 lists the XAUI PHY parameters and the corresponding ALTGX megafunction parameters.

Table 11-2. Comparison of ALTGX Megafunction and XAUI PHY Parameters (Part 1 of 2)

ALTGX Parameter Name (Default Value)	XAUI PHY Parameter Name	Comments
Number of channels	Number of XAUI interfaces	In Stratix V devices, this parameter is locked to 1 (for 4 channels). You cannot change it in the current release.
Train receiver clock and data recover (CDR) from pll_inclk (On)	Not available as parameters in the MegaWizard interface	Use assignment editor to make these assignment
TX PLL bandwidth mode (Auto)		
RX CDR bandwidth mode (Auto)		

Table 11–2. Comparison of ALTGX Megafunction and XAUI PHY Parameters (Part 2 of 2)

ALTGX Parameter Name (Default Value)	XAUI PHY Parameter Name	Comments
Acceptable PPM threshold between receiver CDR VCO and receiver input reference clock (± 1000)	Not available as parameters in the MegaWizard interface	Use assignment editor to make these assignments
Analog power (Auto)		
Loopback option (No loopback)		
Enable static equalizer control (Off)		
DC gain (0)		
Receiver common mode voltage (0.82v)		
Use external receiver termination (Off)		
Receiver termination resistance (100 ohms)		
Transmitter buffer power (1.5v)		
Transmitter common mode voltage (0.65v)		
Use external transmitter termination (Off)		
Transmitter termination resistance (100 ohms)		
VOD setting (4)		
Preemphasis 1 st post-tap (0)		
Preemphasis pre-tap setting (0)		
Preemphasis second post-tap setting (0)		
Analog controls (Off)		
Enable ADCE (Off)	Not available as parameters in the MegaWizard interface	Not available in 10.0
Enable channel and transmitter PLL reconfig (Off)		
Starting channel number (0)	No longer required	Automatically set to 0. The Quartus II software handles lane assignments
Enable run length violation checking with run length of (40)	Not available as parameters in the MegaWizard interface	Use assignment editor
Enable transmitter bit reversal (Off)		
Word alignment pattern length (10)		

Port Differences

Table 11–5 lists the differences between the top-level signals in Stratix IV GX and Stratix V GX/GS devices.

Table 11–3. Correspondences between XAUI PHY Stratix IV GX and Stratix V Device Signals (Part 1 of 3)⁽¹⁾

Stratix IV GX Devices		Stratix V Devices	
Signal Name	Width	Signal Name	Width
Reference Clocks and Resets			
pll_inclk	1	refclk	1
rx_crucclk	[<n> - 1:0]	Not available	—
coreclkout	1	xgmii_rx_clk	1
rx_coreclk	[<n> - 1:0]	Not available	—

Table 11–3. Correspondences between XAUI PHY Stratix IV GX and Stratix V Device Signals (Part 2 of 3)⁽¹⁾

Stratix IV GX Devices		Stratix V Devices	
Signal Name	Width	Signal Name	Width
tx_coreclk	[<n> – 1:0]	xgmii_tx_clk	1
Not available	—	rx_pma_ready	1
Not available	—	tx_pma_ready	1
Data Ports			
rx_datain	[<n>-1:0]	xaui_rx_serial	[3:0]
tx_datain	[16<n>-1:0]	xgmii_tx_dc	[63:0]
rx_dataout	[16<n> – 1:0]	xgmii_rx_dc	[63:0]
tx_dataout	[<n> -1:0]	xaui_tx_serial	[3:0]
Optional TX and RX Status Ports			
gxb_powerdown	[<n>/4 – 1:0]	Not available, however you can access them through the Avalon-MM PHY management interface.	—
pll_locked	[<n>-1:0]	Not available	—
rx_locktorefclk	[<n> -1:0]	Not available	—
rx_locktodata	[<n> -1:0]	Not available	—
rx_pll_locked	[<n>/4 – 1:0]	Not available	—
rx_freqlocked	[<n>/4 – 1:0]	Not available	—
rx_phase_comp_fifo_error	[<n>/4 – 1:0]	Not available	—
tx_phase_comp_fifo_error	[<n>/4 – 1:0]	Not available	—
cal_blk_powerdown	—	Not available	—
rx_syncstatus	[2<n> – 1:0]	rx_syncstatus	[<n>*2 – 1:0]
rx_patterndetect	[2<n> – 1:0]	Not available	—
rx_invpolarity	[<n> – 1:0]	Not available	—
rx_ctrldetect	[2<n> – 1:0]	Not available	—
rx_errdetect	[2<n> – 1:0]	rx_errdetect	[<n>*2 – 1:0]
rx_disperr	[2<n> – 1:0]	rx_disperr	[<n>*2 – 1:0]
tx_invpolarity	[<n> – 1:0]	Not available	—
rx_runningdisp	[2<n> – 1:0]	Not available	—
rx_rmfifo full	[2<n> – 1:0]	Not available	—
rx_rmfifo empty	[2<n> – 1:0]	Not available	—
rx_rmfi fodata inserted	[2<n> – 1:0]	Not available	—
rx_rmfi fodata deleted	[2<n> – 1:0]	Not available	—
Transceiver Reconfiguration			
cal_blk_clk	1	These signals are included in the reconfig_to_xcvr bus.	—
reconfig_clk	1		—
reconfig_togxb	[3:0]	reconfig_to_xcvr	variable
reconfig_fromgxb	[16:0]	reconfig_from_xcvr	variable
Avalon MM Management Interface			

Table 11-3. Correspondences between XAUI PHY Stratix IV GX and Stratix V Device Signals (Part 3 of 3)⁽¹⁾

Stratix IV GX Devices		Stratix V Devices	
Signal Name	Width	Signal Name	Width
Not available		phy_mgmt_clk_rst	1
		phy_mgmt_clk	1
		phy_mgmt_address	[8:0]
		phy_mgmt_read	1
		phy_mgmt_readdata	[31:0]
		phy_mgmt_write	1
		phy_mgmt_writedata	[31:0]

Note to Table 11-3:

(1) <n> = the number of lanes. <d> = the total deserialization factor from the pin to the FPGA fabric.

PHY IP Core for PCI Express PHY (PIPE)

This section lists the differences between the parameters and signals for the PCI Express PHY (PIPE) IP core and the ALTGX megafunction when configured in the PCI Express (PIPE) functional mode.

Parameter Differences

Table 11-4 lists the PHY IP core for PCI Express PHY (PIPE) parameters and the corresponding ALTGX megafunction parameters.

Table 11-4. Comparison of ALTGX Megafunction and PHY IP Core for PCI Express PHY (PIPE) Parameters (Part 1 of 2)

ALTGX Parameter Name (Default Value)	PCI Express PHY (PIPE) Parameter Name	Comments
Number of channels	Number of Lanes	—
Channel width	Deserialization factor	—
Subprotocol	Protocol Version	—
input clock frequency	PLL reference clock frequency	—
Starting Channel Number	—	Automatically set to 0. Quartus II software handles lane assignments.
Enable low latency sync	pipe_low_latency_synchronous_mode	—
Enable RLV with run length of	pipe_run_length_violation_checking	Always on
Enable electrical idle inference functionality	Enable electrical idle inferencing	—
—	phy_mgmt_clk_in_mhz	For embedded reset controller to calculate delays

Table 11-4. Comparison of ALTGX Megafunction and PHY IP Core for PCI Express PHY (PIPE) Parameters (Part 2 of 2)

ALTGX Parameter Name (Default Value)	PCI Express PHY (PIPE) Parameter Name	Comments
Train receiver CDR from pll_inclk (false)	Not available in MegaWizard Interface	Use assignment editor to make these assignments
TX PLL bandwidth mode (Auto)		
RX CDR bandwidth mode (Auto)		
Acceptable PPM threshold (± 300)		
Analog Power(VCCA_L/R) (Auto)		
Reverse loopback option (No loopback)		
Enable static equalizer control (false)		
DC gain (1)		
RX Vcm (0.82)		
Force signal detection (Off)		
Signal Detect threshold (4)		
Use external receiver termination (Off)		
RX term (100)		
Transmitter buffer power(VCCH) (1.5)		
TX Vcm (0.65)		
Use external transmitter termination (Off)		
TX Rterm (100)	Not available in MegaWizard Interface	Use assignment editor to make these assignments
VCO control setting (5)		
Pre-emphasis 1st post tap (18)		
Pre-tap (0)		
2nd post tap (0)		
DPRIO - V _{OD} , Pre-em, Eq and EyeQ (Off)	Not available in MegaWizard Interface	Use assignment editor to make these assignments
DPRIO - Channel and TX PLL Reconfig (Off)		

Port Differences

Table 11-5 lists the differences between the top-level signals in Stratix IV GX and Stratix V GX/GS devices. PIPE standard ports remain, but are now prefixed with pipe_. Clocking options are simplified to match the PIPE 2.0 specification.

Table 11-5. PCIe PHY (PIPE) Correspondence between Stratix IV GX Device and Stratix V Device Signals (Part 1 of 3) ⁽¹⁾

Stratix IV GX Device Signal Name	Stratix V Device Signal Name	Width
Reference Clocks and Resets		
pll_inclk	pll_ref_clk	1
rx_crucclk	Not available	[<n>-1:0]
tx_coreclk	Not available	[<n>-1:0]
rx_coreclk	Not available	[<n>-1:0]
tx_clkout/coreclkout	pipe_pclk	1

Table 11-5. PCIe PHY (PIPE) Correspondence between Stratix IV GX Device and Stratix V Device Signals (Part 2 of 3) ⁽¹⁾

Stratix IV GX Device Signal Name	Stratix V Device Signal Name	Width
pll_powerdown	These signals are now available as control and status registers. Refer to the “Avalon-MM PHY Management Interface” on page 6-14 and “PCI Express PHY (PIPE) IP Core Registers” on page 6-14	1
cal_blk_powerdown		1
Not available	tx_ready (reset control status)	1
Not available	rx_ready (reset curl status)	1
PIPE interface Ports		
tx_datain	pipe_txdata	[<n><d>-1:0]
tx_ctrlenable	pipe_txdatak	[(<d>/8)*<n>-1:0]
tx_detectrxloop	pipe_txdetectrx_loopback	[<n>-1:0]
tx_forcedispcompliance	pipe_txcompliance	[<n>-1:0]
tx_forceelecidle	pipe_txelecidle	[<n>-1:0]
txswing	pipe_txswing	[<n>-1:0]
tx_pipedeemph[0]	pipe_txdeemph	[<n>-1:0]
tx_pipemargin[2:0]	pipe_txmargin	[3<n>-1:0]
rateswitch[0]	pipe_rate[1:0]	[<n>-1:0]
powerdn	pipe_powerdown	[2<n>-1:0]
rx_elecidleinfersel	pipe_eidleinfersel	[3<n>-1:0]
rx_dataout	pipe_rxdata	[<n>-*<d>-1:0]
rx_ctrldetect	pipe_rxdatak	[(<d>/8)*<n>-1:0]
pipedatavalid	pipe_rxvalid	[<n>-1:0]
pipe8b10binvpolarity	pipe_rxpolarity	[<n>-1:0]
pipeelecidle	pipe_rxelecidle	[<n>-1:0]
pipephydonestatus	pipe_phystatus	[<n>-1:0]
pipestatus	pipe_rxstatus	[3<n>-1:0]
Non-PIPE Ports		
rx_pll_locked	rx_is_lockedtoref	[<n>--1:0]
rx_freqlocked	rx_is_lockedtodata	[<n>--1:0]
pll_locked	pll_locked	1
rx_syncstatus	rx_syncstatus (also management interface)	[(<d>/8)*<n>-1:0]

Table 11-5. PCIe PHY (PIPE) Correspondence between Stratix IV GX Device and Stratix V Device Signals (Part 3 of 3) ⁽¹⁾

Stratix IV GX Device Signal Name	Stratix V Device Signal Name	Width
rx_locktodata	These signals are now available as control and status registers. Refer to the “ Register Descriptions ” under “ Avalon-MM PHY Management Interface ” on page 6-14	[<n>-1:0]
rx_locktorefclk		[<n>-1:0]
tx_invpolarity		[<n>-1:0]
rx_errdetect		[(<d>/8)*<n>-1:0]
rx_disperr		[(<d>/8)*<n>-1:0]
rx_patterndetect		[(<d>/8)*<n>-1:0]
tx_phase_comp_fifo_error		[<n>-1:0]
rx_phase_comp_fifo_error		[<n>-1:0]
rx_signaldetect		[<n>-1:0]
rx_rlv		[<n>-1:0]
rx_datain	rx_serial_data	[<n>-1:0]
tx_dataout	tx_serial_data	[<n>-1:0]
Reconfiguration		
cal_blk_clk	These signals are included in the reconfig_to_xcvr bus	1
reconfig_clk		1
fixedclk		1
reconfig_togxb	reconfig_to_xcvr	variable
reconfig_fromgxb	reconfig_from_xcvr	variable
Avalon MM Management Interface		
Not available	phy_mgmt_clk_reset	1
	phy_mgmt_clk	1
	phy_mgmt_address	[8:0]
	phy_mgmt_read	1
	phy_mgmt_readdata	[31:0]
	phy_mgmt_write	1
	phy_mgmt_writedata	[31:0]

Note to Table 11-5:

(1) <n> = the number of lanes. <d> = the total deserialization factor from the pin to the FPGA fabric.

Custom PHY

This section lists the differences between the parameters and signals for the Custom PHY IP core and the ALTGX megafunction when configured in the Basic functional mode.

Parameter Differences

Table 11-6 lists the Custom PHY parameters and the corresponding ALTGX megafunction parameters.

Table 11-6. Comparison of ALTGX Megafunction and Custom PHY Parameters (Part 1 of 2)

ALTGX Parameter Name (Default Value)	Custom PHY Parameter Name
General	
Not available	Device family
	Transceiver protocol
	Mode of operation
	Enable bonding
What is the number of channels?	Number of lanes
Which subprotocol will you be using? (x4, x8)	Not available
What is the channel width?	Serialization factor
What is the effective data rate?	Data rate
What is the input clock frequency?	Input clock frequency
tx/rx_8b_10b_mode	Enable 8B/10B encoder/decoder
Not available	Enable manual disparity control
	Create optional 8B10B status ports
What is the deserializer block width?	Deserializer block width: ⁽¹⁾
Single Double	Auto Single Double
Additional Options	
Not available	Enable TX Bitslip
	Create rx_coreclkkin port
	Create tx_coreclkkin port
	Create rx_recovered_clk port
	Create optional ports
	Avalon data interfaces
	Force manual reset control
Protocol Settings-Word Aligner	Word Aligner
Use manual word alignment mode Use manual bitslipping mode Use the built-in 'synchronization state machine'	Word alignment mode
Enable run length violation checking with a run length of	Run length
What is the word alignment pattern	Word alignment pattern

Table 11-6. Comparison of ALTGX Megafunction and Custom PHY Parameters (Part 2 of 2)

What is the word alignment pattern length	Word aligner pattern length
Protocol Settings—Rate match/Byte order	Rate Match
What is the 20-bit rate match pattern1 (usually used for +ve disparity pattern)	Rate match insertion/deletion +ve disparity pattern
What is the 20-bit rate match pattern1 (usually used for -ve disparity pattern)	Rate match insertion/deletion -ve disparity pattern
Protocol Settings—Rate match/Byte order	Byte Order
What is the byte ordering pattern	Byte ordering pattern

Note to Table 11-6:

(1) This parameter is on the Datapath tab.

Port Differences

Table 11-5 lists the differences between the top-level signals in Stratix IV GX and Stratix V GX/GS devices.

Table 11-7. Custom PHY Correspondences between Stratix IV GX Device and Stratix V Device Signals

ALTGX	Custom PHY	Width
Avalon MM Management Interface		
Not available	phy_mgmt_clk_reset	1
	phy_mgmt_clk	1
	phy_mgmt_address	8
	phy_mgmt_read	1
	phy_mgmt_readdata	32
	phy_mgmt_write	1
	phy_mgmt_writedata	32
Clocks		
cal_blk_clk	These signals are included in the reconfig_to_xcvr bus	
reconfig_clk		
p1l_inclk	p1l_ref_clk	[<p>-1:0]
rx_coreclk	rx_coreclkin	
tx_coreclk	tx_coreclkin	
Avalon-ST TX Interface		
tx_datain	tx_parallel_data	[<d><n>-1:0]
tx_ctrlenable	tx_dataak	[<d><n>-1:0]
rx_ctrldetect	rx_dataak	[<d><n>-1:0]
Avalon-ST RX Interface		
rx_dataout	rx_parallel_data	[<d><n>-1:0]
rx_runningdisp	rx_runningdisp	[<d/8><n>-1:0]
rx_enabyteord	rx_enabyteord	[<n>-1:0]

Table 11-7. Custom PHY Correspondences between Stratix IV GX Device and Stratix V Device Signals

High Speed Serial I/O		
rx_datain	rx_serial_data	[<n>-1:0]
tx_dataout	tx_serial_data	[<n>-1:0]
rx_freqlocked	rx_is_lockedtodata	[<n>-1:0]
Transceiver Control and Status Signals		
gxb_powerdown	phy_mgmt_clk_reset	
rx_dataoutfull	—	
tx_dataoutfull	—	
rx_pll_locked	There are both pll_locked and rx_pll_clocked in Stratix IV. Stratix V only has pll_locked.	
rx_clkout	These signals are now available as control and status registers. Refer to “Register Descriptions” on page 7–21.	
rx_phase_comp_fifo_error		
rx_serialpbken		
tx_phase_comp_fifo_error		
tx_invpolarity		
Transceiver Reconfiguration		
reconfig_togxb[3:0]	reconfig_to_xcvr	variable
reconfig_fromgxb[16:0]	reconfig_from_xcvr	variable

Note to Table 11-7:

(1) <n> = the number of lanes. <d> = the total deserialization factor from the pin to the FPGA fabric.

This chapter provides additional information about the document and Altera.

Revision History

The table below displays the revision history for the chapters in this user guide.

Date	Version	Changes Made
Custom		
March 2012	1.6	■ Added register definitions for address range 0x080–0x085. .
Low Latency PHY		
March 2012	1.6	■ Removed register definitions for address range 0x080–0x085. .
10GBASE-R		
February 2012	1.5	■ Added datapath latency numbers for Stratix V devices. ■ Corrected bit range for <code>ERRORED_BLOCK_COUNT</code>. ■ Added statement that the <code>cal_blk_powerdown</code> (0x021) and <code>pma_tx_pll_is_locked</code> (0x022) registers are only available when the Use external PMA control and reconfig option is turned On on the Additional Options tab of the GUI. ■ Clarified that the BER count functionality is for Stratix IV devices only. ■ Removed <code>pma_rx_signaldetect</code> register. The 10GBASE-R PHY does not support this functionality.
XAUI		
February 2012	1.5	■ Removed reset bits at register 0x081. The reset implemented Cat register 0x044 provides more comprehensive functionality. ■ Removed <code>pma_rx_signaldetect</code> register. The XAUI PHY does not support this functionality.
PCI Express (PIPE)		
February 2012	1.5	■ Updated definition of <code>fixedclk</code> . It can be derived from <code>pll_ref_clk</code> .
Custom		
February 2012	1.5	■ Removed register definitions for Low Latency PHY.
Low Latency PHY		
February 2012	1.5	■ Added register definitions for Low Latency PHY.
Deterministic Latency PHY		
February 2012	1.5	■ Removed <code>pma_rx_signaldetect</code> register. The Deterministic Latency PHY does not support this functionality. ■ Updated the definition of deterministic latency word alignment mode to include the fact that the word alignment pattern, which is currently forced to K28.5 = 0011111010 is always placed in the least significant byte (LSB) of a word with a fixed latency of 3 cycles.

Date	Version	Changes Made
Transceiver Reconfiguration Controller		
February 2012	1.5	Added DFE.
Introduction		
December 2011	1.4	Revised discussion of embedded reset controller to include the fact that this reset controller can be disabled for some transceiver PHYs.
10GBASE-R		
December 2011	1.4	- Removed description of calibration block powerdown register (0x021) which is not available for this transceiver PHY. - Changed definition of <code>phy_mgmt_clk_reset</code>. This signal is active high and level sensitive.
XAUI		
December 2011	1.4	- Changed definition of <code>phy_mgmt_clk_reset</code>. This signal is active high and level sensitive. - Added Arria II GX to device support table.
Interlaken		
December 2011	1.4	- Changed access mode for RX equalization, pre-CDR reverse serial loopback, and post-CDR reverse serial loopback to write only (WO). - Removed optional <code>rx_sync_word_err</code>, <code>rx_scrm_err</code>, and <code>rx_framing_err</code> status bits. - Changed definition of <code>phy_mgmt_clk_reset</code>. This signal is active high and level sensitive.
PHY IP Core for PCI Express (PIPE)		
December 2011	1.4	Changed definition of <code>phy_mgmt_clk_reset</code>. This signal is active high and level sensitive.
Custom		
December 2011	1.4	- Added $\times N$ and feedback compensation options for bonded clocks. - Added Enable Channel Interface parameter which is required for dynamic reconfiguration of transceivers. - Corrected formulas for signal width in top-level signals figure. - Changed definition of <code>phy_mgmt_clk_reset</code>. This signal is active high and level sensitive.
Low Latency PHY		
December 2011	1.4	- Added option to disable the embedded reset controller to allow you to create your own reset sequence. - Added $\times N$ and feedback compensation options for bonded clocks. - Fixed name of <code>phy_mgmt_reset</code> signal. Should be <code>phy_mgmt_clk_reset</code>. Also, a positive edge on this signal initiates a reset. - Added Enable Channel Interface parameter which is required for dynamic reconfiguration of transceivers. - Corrected formulas for signal width in top-level signals figure. - Changed definition of <code>phy_mgmt_clk_reset</code>. This signal is active high and level sensitive.
Deterministic Latency PHY		
December 2011	1.4	- Removed Enable tx_clkout feedback path for TX PLL from the General Options tab of the Deterministic Latency PHY IP Core GUI. This option is unavailable in 11.1 and 11.1 SP1. - Changed definition of <code>phy_mgmt_clk_reset</code>. This signal is active high and level sensitive.

Date	Version	Changes Made
Transceiver Reconfiguration Controller		
December 2011	1.4	■ Added duty cycle distortion (DCD) signal integrity feature. ■ Added PLL and channel reconfiguration using a memory initialization file (.mif). ■ Added ability to reconfigure PLLs, including the input reference clock or to change the PLL that supplies the high speed serial clock to the serializer without including logic to reconfigure channels. ■ Corrected values for RX equalization gain. 0–4 are available. ■ Corrected logical number in “Interface Ordering with Multiple Transceiver PHY Instances” on page 10–34. ■ Increased the number of channels that can share a PLL from 5 to 11 when feedback compensation is used. ■ Increased the number of channels that can connect to the Transceiver Reconfiguration Controller from 32 to 64. ■ Added section on requirements for merging PLLs.
Introduction		
November 2011	1.3	■ Revised reset section. The 2 options for reset are now the embedded reset controller or user-specified reset controller. ■ Updated directory names in simulation testbench.
10GBASE-R PHY Transceiver		
November 2011	1.3	■ Added support for Stratix V devices. ■ Added section discussing transceiver reconfiguration in Stratix V devices. ■ Removed rx_oc_busy signal which is included in the reconfiguration bus. ■ Updated QSF settings to include text strings used to assign values and location of the assignment which is either a pin or PLL.
XAUI Transceiver PHY		
November 2011	1.3	■ The pma_tx_pll_is_locked is not available in Stratix V devices. ■ Added base data rate, lane rate, input clock frequency, and PLL type parameters. ■ Updated QSF settings to include text strings used to assign values and location of the assignment which is either a pin or PLL. ■ Added section on dynamic transceiver reconfiguration in Stratix V devices. ■ Removed Timing Constraints section. These constraints are included in the HDL code.
Interlaken Transceiver PHY		
November 2011	1.3	■ Added tx_sync_done signal which indicates that all lanes of TX data are synchronized. ■ tx_coreclk_in is required in this release. ■ Added base data rate, lane rate, input clock frequency, and PLL type parameters. ■ Updated QSF settings to include text strings used to assign values and location of the assignment which is either a pin or PLL.
PHY IP Core for PCI Express (PIPE)		

Date	Version	Changes Made
November 2011	1.3	- Added <code>pll_powerdown</code> bit (bit[0] of 0x044) for manual reset control. You must assert this bit for 1 μs for Gen 2 operation. - Added PLL type and base data rate parameters. - Updated QSF settings to include text strings used to assign values and location of the assignment which is either a pin or PLL.
Custom Transceiver PHY		
November 2011	1.3	- Added Arria V and Cyclone V support. - Added base data rate, lane rate, input clock frequency, and PLL type parameters. - Revised reset options. The 2 options for reset are now the embedded reset controller or a user-specified reset logic. - Updated QSF settings to include text strings used to assign values and location of the assignment which is either a pin or PLL.
Low Latency PHY		
November 2011	1.3	- Added base data rate, lane rate, input clock frequency, and PLL type parameters. - Updated QSF settings to include text strings used to assign values and location of the assignment which is either a pin or PLL. - Revised reset options. The 2 options for reset are now the embedded reset controller or a user-specified reset logic.
Deterministic Latency		
November 2011	1.3	Initial release of this chapter.
Transceiver Reconfiguration Controller		
November 2011	1.3	- Added MIF support to allow transceiver reconfiguration from a <code>.mif</code> file that may contain updates to multiple settings. - Added support for the following features: - EyeQ - AEQ - AEQ - ATX tuning - PLL reconfiguration - DC gain and four-stage linear equalization for the RX channels - Removed Stratix IV device support. - Changed frequency range of <code>phy_mgmt_clk</code> to 100-125 MHz.

Date	Version	Changes Made
All Chapters		
July 2011	1.2.1	■ Restricted frequency range of the <code>phy_mgmt_clk</code> to 90–100 MHz for the <i>Transceiver Reconfiguration Controller IP Core</i> chapter. There is no restriction on the frequency of <code>phy_mgmt_clk</code> for Stratix V devices in the 10GBASE-R, XAUI, Interlaken, PHY IP Core for PCI Express, Custom, and Low Latency PHYs; however, to use the same clock source for both, you must restrict this clock to 90–100 MHz. ■ Added column specifying availability of read and write access for PMA analog controls in the <i>Transceiver Reconfiguration Controller IP Core</i> chapter. ■ Renamed Avalon-MM bus in for Transceiver Reconfiguration Controller <code>reconfig_mgmt*</code>. ■ Provided frequency range for <code>phy_mgmt_clk</code> for the XAUI PHY IP Core in Arria II GX, Cyclone IV GX, HardCopy IV, and Stratix IV GX devices. ■ Added register descriptions for the automatic reset controller to the <i>Low Latency PHY IP Core</i> chapter. ■ Added two steps to procedure to reconfigure a PMA control in the <i>Transceiver Reconfiguration Controller</i> chapter. ■ Corrected RX equalization DC gain in <i>transceiver Reconfiguration Controller</i> chapter. It should be 0–4. ■ Corrected serialization factor column in <i>Low Latency PHY IP Core</i> chapter.
Introduction		
May 2011	1.2	■ Added simulation section. ■ Revised Figure 1–1 on page 1–2 to show the Transceiver Reconfiguration Controller as a separately instantiated IP core. ■ Added statement saying that the transceiver PHY IP cores do not support the NativeLink feature of the Quartus II software. ■ Revised reset section.
Getting Started		
May 2011	1.2	■ No changes from previous release.
10GBASE-R PHY Transceiver		
May 2011	1.2	■ Corrected frequency of <code>p11_ref_clk</code>. Should be 644.53125 MHz, not 644.53725 MHz. ■ Renamed <code>reconfig_fromgxb</code> and <code>reconfig_togxb</code> <code>reconfig_from_xcvr</code> and <code>reconfig_to_xcvr</code>, respectively.
XAUI PHY Transceiver		

Date	Version	Changes Made
May 2011	1.2	■ Added support for DDR XAUI ■ Added support for Arria II GZ and HardCopy IV ■ Added example testbench ■ Renamed <code>reconfig_fromgxb</code> and <code>reconfig_togxb</code> to <code>reconfig_from_xcvr</code> and <code>reconfig_to_xcvr</code>, respectively. ■ Updated definitions of <code>rx_digital_reset</code> and <code>tx_digital_reset</code> for the soft XAUI implementation in Table 4-17 on page 4-19. ■ Changed description of <code>rx_syncstatus</code> register and signals to specify 2 bits per channel in hard XAUI and 1 bit per channel in soft XAUI implementations. ■ Corrected bit sequencing for 0x084, 0x085 and 0x088 in Table 4-17 on page 4-19, as follows: ■ <code>patterndetect</code> = 0x084, bits [15:8] ■ <code>syncstatus</code> = 0x084, bits [7:0] ■ <code>errordetect</code> = 0x085, bits [15:8] ■ <code>disperr</code> = 0x085, bits [7:0] ■ <code>rmfifoempty</code> = 0x088, bits [7:4] ■ <code>rmfifoempty</code> = 0x088, bits [3:0]
Interlaken PHY Transceiver		
May 2011	1.2	■ Added details about the 0 ready latency for <code>tx_ready</code>. ■ Added PLL support to lane rate parameter description in Table 5-2 on page 5-2. ■ Moved dynamic reconfiguration for the transceiver outside of the Interlaken PHY IP Core. The reconfiguration signals now connect to a separate Reconfiguration Controller IP Core. ■ Added a reference to PHY IP Design Flow with Interlaken for Stratix V Devices which is a reference design that implements the Interlaken protocol in a Stratix V device. ■ Changed supported metaframe lengths from 1–8191 to 5–8191. ■ Added <code>pll_locked</code> output port. ■ Added <code>indirect_addr</code> register at 0x080 for use in accessing PCS control and status registers. ■ Added new Bonded group size parameter.
PHY IP Core for PCI Express PHY (PIPE)		
May 2011	1.2	■ Renamed to PHY IP Core for PCI Express. ■ Moved dynamic reconfiguration for the transceiver outside of the PHY IP Core. The reconfiguration signals now connect to a separate Reconfiguration Controller IP Core. ■ Removed $\times 2$ support.

Date	Version	Changes Made
Custom PHY Transceiver		
May 2011	1.2	■ Added presets for the 3.25GbE and 1.25GbE protocols. ■ Moved dynamic reconfiguration for the transceiver outside of the Custom PHY IP Core. The reconfiguration signals now connect to a separate Reconfiguration Controller IP Core. ■ Removed device support for Arria II GX, Arria II GZ, HardCopy IV GX, and Stratix IV GX. ■ Added the following parameters on the General tab: ■ Transceiver protocol ■ Create rx_recovered_clk port ■ Force manual reset control ■ Added optional rx_rmifodddatainserted, rx_rmifodatadelted, rx_rlv, and rx_recovered_clk as output signals. ■ Added phy_mgmt_waitrequest to the PHY management interface. ■ Renamed reconfig_fromgxb and reconfig_togxb reconfig_from_xcvr and reconfig_to_xcvr, respectively. ■ Corrected address for 8-Gbps RX PCS status register in Table 7-21 on page 7-21. ■ Added special pad requirement for Byte ordering pattern. Refer to Table 7-7 on page 7-7. ■ Clarified behavior of the word alignment mode. Added note explaining how to disable all word alignment functionality.
Low Latency PHY Transceiver		
May 2011	1.2	■ Moved dynamic reconfiguration for the transceiver outside of the Low Latency PHY IP Core. The reconfiguration signals now connect to a separate Reconfiguration Controller IP Core. ■ Moved dynamics reconfiguration for the transceiver outside of the Custom PHY IP Core. The reconfiguration signals now connect to a separate Reconfiguration Controller IP Core. ■ Renamed the tx_parallel_clk signal tx_clkout.
Transceiver Reconfiguration Controller		
May 2011	1.2	■ Added Stratix V support. The Transceiver Reconfiguration Controller is only available for Stratix IV devices in the Transceiver Toolkit. ■ Added sections describing the number of reconfiguration interfaces required and restrictions on channel placement. ■ Added pre- and post-serial loopback controls. ■ Changed reconfiguration clock source. In 10.1, the Avalon-MM PHY Management clock was used for reconfiguration. In 11.0, the reconfiguration controller supplies this clock.
Migrating from Stratix IV to Stratix V		
May 2011	1.2	■ Added discussion of dynamic reconfiguration for Stratix IV and Stratix V devices. ■ Added information on loopback modes for Stratix IV and Stratix V devices. ■ Added new parameters for Custom PHY IP core in Stratix V devices.

Date	Version	Changes Made
All Chapters		
December 2010	1.11	■ Corrected frequency range for the <code>phy_mgmt_clk</code> for the Custom PHY IP core in Table 7–20 on page 7–21. ■ Added optional <code>reconfig_from_xcvr[67:0]</code> to Figure 4–4 on page 4–12. Provided more detail on size of <code>reconfig_from_xcvr</code> in Table 4–18 on page 4–23 ■ Removed table providing ordering codes for the Interlaken PHY IP core. Ordering codes are not required for Stratix V devices using the hard implementation of the Interlaken PHY. ■ Added note to 10GBASE-R release information table stating that “No ordering codes or license files are required for Stratix V devices.” ■ Minor update to the steps to reconfigure a TX or RX PMA setting in the Transceiver Reconfiguration Controller chapter.
Introduction		
December 2010	1.1	■ Revised reset diagram. ■ Added block diagram for reset ■ Removed support for SOPC Builder
Getting Started		
December 2010	1.1	■ Removed description of SOPC Builder design flow. SOPC Builder is not supported in this release.
10GBASE-R PHY Transceiver		
December 2010	1.1	■ Added Stratix V support ■ Changed <code>phy_mgmt_address</code> from 16 to 9 bits. ■ Renamed management interface, adding <code>phy_</code> prefix ■ Renamed <code>block_lock</code> and <code>hi_ber</code> signals <code>rx_block_lock</code> and <code>rx_hi_ber</code>, respectively. ■ Added top-level signals for external PMA and reconfiguration controller in Stratix IV devices. Refer to Table 3–18 on page 3–20. ■ Removed the <code>mgmt_burstcount</code> signal. ■ Changed register map to show word addresses instead of a byte offset from a base address.
XAUI PHY Transceiver		
December 2010	1.1	■ Added support for Arria II GX and Cyclone IV GX with hard PCS ■ Renamed management interface, adding <code>phy_</code> prefix ■ Changed <code>phy_mgmt_address</code> from 16 to 9 bits. ■ Renamed many signals. Refer to “XAUI Top-Level Signals—Soft PCS and PMA” on page 4–12 and “XAUI Top-Level Signals—Hard IP PCS and PMA” on page 4–11 as appropriate. ■ Changed register map to show word addresses instead of a byte offset from a base address. ■ Removed the <code>rx_ctrldetect</code> and <code>rx_freqlocked</code> signals.

Date	Version	Changes Made
Interlaken PHY Transceiver		
December 2010	1.1	■ Added simulation support in ModelSim SE, Synopsys VCS MX, Cadence NCSim ■ Changed number of lanes supported from 4–24 to 1–24. ■ Changed reference clock to be 1/20th rather than 1/10th the lane rate. ■ Renamed management interface, adding <code>phy_</code> prefix ■ Changed <code>phy_mgmt_address</code> from 16 to 9 bits. ■ Changed many signal names, refer to Figure 5–2 on page 5–8. Changed register map to show word addresses instead of a byte offset from a base address.
PCI Express PHY (PIPE)		
December 2010	1.1	■ Added simulation support in ModelSim SE ■ Added PIPE low latency configuration option ■ Changed <code>phy_mgmt_address</code> from 16 to 9 bits. ■ Changed register map to show word addresses instead of a byte offset from a base address. ■ Added <code>tx_ready</code>, <code>rx_ready</code>, <code>pipe_txswing</code>, and <code>pipe_rxeleciidle</code> signals ■ Added <code>rx_errdetect</code>, <code>rx_disperr</code>, and <code>rx_ala2sizeout</code> register fields
Custom PHY Transceiver		
December 2010	1.1	■ Added support for 8B/10B encoding and decoding in Stratix V devices ■ Added support for rate matching in Stratix V devices. ■ Added support for Arria II GX, Arria II GZ, HardCopy IV GX, and Stratix IV GX devices ■ Renamed management interface, adding <code>phy_</code> prefix ■ Changed <code>phy_mgmt_address</code> from 8 to 9 bits. ■ Added many optional status ports and renamed some signals. Refer to Figure 7–2 on page 7–16 and subsequent signal descriptions. ■ Changed register map to show word addresses instead of a byte offset from a base address.
Low Latency PHY IP Core		
December 2010	1.1	■ Renamed management interface, adding <code>phy_</code> prefix ■ Changed <code>phy_mgmt_address</code> from 16 to 9 bits. ■ Changed register map to show word addresses instead of a byte offset from a base address. ■ Removed <code>rx_offset_cancellation_done</code> signal. Internal reset logic determines when offset cancellation has completed. ■ Removed support for Stratix IV GX devices.
Transceiver Reconfiguration Controller		
December 2010	1.1	■ Reconfiguration is now integrated into the XAUI PHY IP core and 10GBASE-R PHY IP core. ■ Revised register map to show word addresses instead of a byte offset from a base address.
Migrating from Stratix IV to Stratix V		
December 2010	1.1	■ Changed <code>phy_mgmt_address</code> from 16 to 9 bits.

Date	Version	Changes Made
November 2010	1.1	- Corrected address offsets in Table 10-9 on page 10-11. These are byte offsets and should be: 0x00, 0x04, 0x08, 0x0C, 0x10, not 0x00, 0x01, 0x02, 0x03, 0x04. - Corrected base address for transceiver reconfiguration control and status registers in Table 10-9 on page 10-11. It should be 0x420, not 0x400. - Corrected byte offsets in Table 7-21 on page 7-21 and Table 6-12 on page 6-14. The base address is 0x200. The offsets are 0x000–0x018.
July 2010	1.0	Initial release.

How to Contact Altera

To locate the most up-to-date information about Altera products, refer to the following table.

Contact ⁽¹⁾	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Product literature	Website	www.altera.com/literature
Nontechnical support (general) (software licensing)	Email	nacomp@altera.com
	Email	authorization@altera.com

Note to Table:

(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

The following table shows the typographic conventions this document uses.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Indicate command names, dialog box titles, dialog box options, and other GUI labels. For example, Save As dialog box. For GUI elements, capitalization matches the GUI.
bold type	Indicates directory names, project names, disk drive names, file names, file name extensions, software utility names, and GUI labels. For example, \qdesigns directory, D: drive, and chiptrip.gdf file.
<i>Italic Type with Initial Capital Letters</i>	Indicate document titles. For example, <i>Stratix IV Design Guidelines</i> .
<i>italic type</i>	Indicates variables. For example, $n + 1$. Variable names are enclosed in angle brackets (< >). For example, <file name> and <project name>.pdf file.
Initial Capital Letters	Indicate keyboard keys and menu names. For example, the Delete key and the Options menu.
“Subheading Title”	Quotation marks indicate references to sections in a document and titles of Quartus II Help topics. For example, “Typographic Conventions.”

Visual Cue	Meaning
Courier type	Indicates signal, port, register, bit, block, and primitive names. For example, <code>data1</code> , <code>tdi</code> , and <code>input</code> . The suffix <code>n</code> denotes an active-low signal. For example, <code>resetn</code> . Indicates command line commands and anything that must be typed exactly as it appears. For example, <code>c:\qdesigns\tutorial\chiptrip.gdf</code> . Also indicates sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword <code>SUBDESIGN</code>), and logic function names (for example, <code>TRI</code>).
	An angled arrow instructs you to press the Enter key.
1., 2., 3., and a., b., c., and so on	Numbered steps indicate a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ■ ■	Bullets indicate a list of items when the sequence of the items is not important.
	The hand points to information that requires special attention.
	The question mark directs you to a software help system with related information.
	The feet direct you to another document or website with related information.
	The multimedia icon directs you to a related multimedia presentation.
	A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.
	A warning calls attention to a condition or possible situation that can cause you injury.
	The envelope links to the Email Subscription Management Center page of the Altera website, where you can sign up to receive update notifications for Altera documents.
	The feedback icon allows you to submit feedback to Altera about the document. Methods for collecting feedback vary as appropriate for each document.

